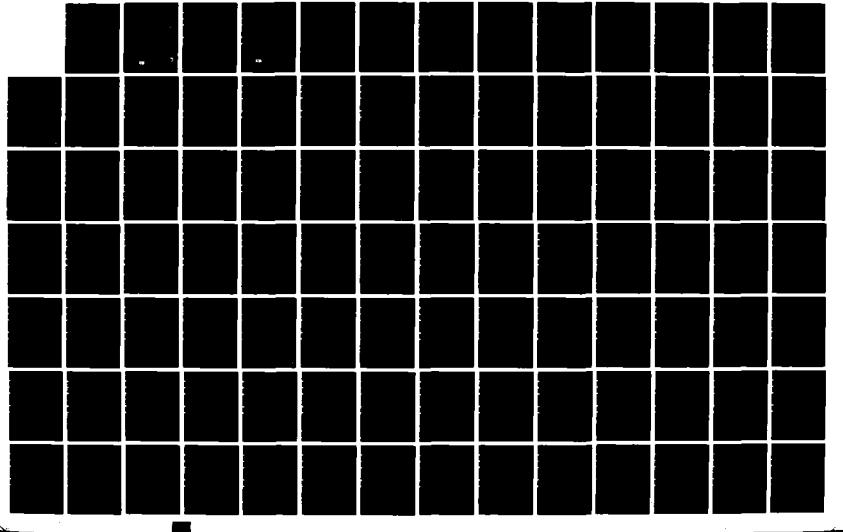
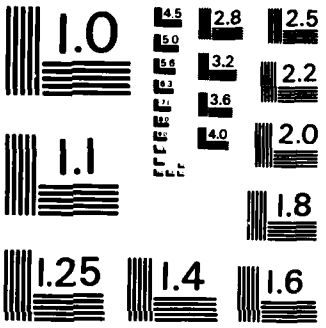
AD-A145 854 A SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM FOR SOLVING LARGE SPARSE NON..(U) CENTER FOR NAVAL ANALYSES ALEXANDRIA VA AIR WARFARE RESEARCH D.. 1/2

UNCLASSIFIED R H NICKEL ET AL. MAY 84 CNA-PP-416 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

PROFESSIONAL PAPER 416 / May 1984

A SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM FOR SOLVING LARGE, SPARSE NONLINEAR PROGRAMS

Ronald H. Nickel
Jon W. Tolle

N00014-83-C-1725

CNA
CENTER FOR NAVAL ANALYSES

The ideas expressed in this paper are those of the authors. The paper does not necessarily represent the views of the Center for Naval Analyses.

PROFESSIONAL PAPER 416 / May 1984

A SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM FOR SOLVING LARGE, SPARSE NONLINEAR PROGRAMS

Ronald H. Nickel
Jon W. Tolle*

*Curriculum in Operations Research and Systems Analysis
University of North Carolina
Chapel Hill, North Carolina 27514

Air Warfare Research Department

CENTER FOR NAVAL ANALYSES

2000 North Beauregard Street, Alexandria, Virginia 22311

ABSTRACT

This document describes

~~Described here is~~ the structure and theory for a sequential quadratic programming algorithm for solving large, sparse nonlinear optimization problems. Also provided are the details of a computer implementation of the algorithm, along with test results. The algorithm is based on Han's sequential quadratic programming method. It maintains a sparse approximation to the Cholesky factor of the Hessian of the Lagrangian and stores all gradients in a sparse format. The solution to the quadratic program generated at each step is obtained by solving the dual quadratic program using a projected conjugate gradient algorithm. Since only active constraints are considered in forming the dual, the dual problem will normally be much smaller than the primal quadratic program and, hence, much easier to solve. An updating procedure is employed that does not destroy sparsity.

Several test problems, ranging in size from 5 to 60 variables were solved with the algorithm. These results indicate that the algorithm has the potential to solve large, sparse nonlinear programs. The algorithm is especially attractive for solving problems having nonlinear constraints.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

INTRODUCTION

Constrained nonlinear optimization has been the subject of a significant amount of research during the past two decades. As a result, a variety of different types of algorithms for solving nonlinear programs have been developed and tested. (See Lasdon and Waren [1979] for a report on the status of nonlinear programming software.) Many of these algorithms, some based on the sequential quadratic programming (SQP) method to be described later, have been found quite efficient for solving small to medium-sized problems. As yet, however, there have been few attempts to construct algorithms for solving large-scale nonlinear programs.

Those algorithms for which software now exists are not readily adapted to large problems because they typically do not take advantage of the sparsity of the Hessian matrices normally associated with large-scale systems. This is a serious defect since the storage and handling of large, dense Hessian matrices is prohibitively expensive. Those few algorithms that have been specifically designed for large-scale problems are normally considered most efficient for special types of problems, such as geometric programs or those with linear constraints (for instance, see Murtagh and Saunders [1982]).

This paper presents an algorithm, using the aforementioned SQP method, that has been developed to handle large nonlinear programs, including those with nonlinear constraints.

This algorithm has several unique features. It maintains a sparse approximation to the Cholesky factorization of the Hessian of the Lagrangian function. The quadratic program generated at each iteration of the SQP method is transformed into a quadratic program having only nonnegativity constraints corresponding to the multipliers associated with inequality constraints in the original nonlinear program. The transformed quadratic program, which is always feasible, is then solved using a projected preconditioned conjugate gradient (CG) method, and finally, the algorithm uses a quasi-Newton update scheme for the factorization of the Hessian approximation that ignores fill-in.

The general form of the nonlinear program to be considered here is

$$(1) \quad \min_{x \in R^n} f(x)$$

$$\text{subject to} \quad g_j(x) \leq 0, \; j = 1, 2, \ldots, m,$$

$$h_k(x) = 0, \; k = m + 1, \ldots, m + p.$$

We will normally assume that f, g_1, g_2, $\ldots$, g_m, h_{m+1}, $\ldots$, h_{m+p} are twice continuously differentiable.

Part I of this report discusses the structure and theory of the algorithm. The discussion focuses on the features that differ from a standard SQP algorithm. Part II describes the actual implementation of the algorithm and presents test results for several problems.

NOTATION

We follow the following notational conventions. The gradient of a real-valued function f of the vector x will be denoted by $\nabla f(x)$, the Hessian of f by $\nabla^2 f(x)$. The multiplier vector will be written as $(u', v')'$, where u is the multiplier vector corresponding to inequality constraints and v is the multiplier vector for equality constraints. The transpose of a matrix Q will be denoted by Q'. Likewise, the transpose of the column vector u will be the row vector u'. Note that no special notation is used for the multipliers corresponding to upper or lower bounds on variables.

The Lagrangian function will be denoted by $\ell(x, u, v)$ with the Hessian of the Lagrangian denoted by $\nabla^2_{xx}\ell(x, u, v)$ and its positive definite representation by $Q=LL'$ where L is a lower triangular matrix.

The step vector will be s and the step length parameter will be α. The current estimate of a solution will be given by x^c and a new estimate by x^n.

For a scalar a, $[a]_+ = \max\{0, a\}$, $|a| = \max\{-a, a\}$. For a vector z, let $|z| = \max\{|z_i|\}$ and $|z|_2 = \left[\Sigma z_i^2 \right]^{1/2}$.

Let A be a matrix. The i, j-th element of A is A_{ij}, the j-th column is $A_{\cdot j}$ and the i-th row is $A_{i\cdot}$. The unit vector having all zeros except in the i-th component will be denoted by e_i.

PART I. STRUCTURE AND THEORY

I.1 THE SEQUENTIAL QUADRATIC PROGRAMMING METHOD

The sequential quadratic programming method generates a sequence of quadratic programs (QP) that approximate the local behavior of the original nonlinear program (1). The Hessian of this subproblem is updated from iteration to iteration using one of the variable metric updating formulas (Han [1976]). The solution of the QP subproblem generated at each iteration determines the step direction for that iteration, and the multiplier vector associated with this solution to the QP is taken as an approximation to the multiplier vector of (1). The following is a brief overview of the SQP method.

Initially, let us assume that (1) has only inequality constraints. The generalization to include equality constraints is straightforward. Let $x^c \in R^n$ be the current estimate of the solution to (1), $u^c \in R^m$ the current estimate of the Lagrangian multiplier vector associated with the solution of (1), and $\ell(x, u)$ the Lagrangian function, i.e.,

$$\ell(x, u) = f(x) + u'g(x).$$

Let (x^*, u^*) be a Kuhn-Tucker point corresponding to a local minimum of (1); that is, (x^*, u^*) satisfies the following:

(2a) $\nabla_x \ell(x^*, u^*) = 0$,

(2b) $u^{*\prime} g(x^*) = 0$,

(2c) $u^* \geq 0$, and

(2d) $g(x^*) \leq 0$.

The SQP algorithm determines $s_x^c \in R^n$ and $s_u^c \in R_m$ such that $(x^c + s_x^c, u^c + s_u^c)$ is a first-order approximate solution to the equation defined by (2a) - (2d). It can be shown that s_x^c is the solution to the following quadratic program:

$$(3) \quad \min_{s_x} \quad \nabla f(x^c)' s_x + (1/2) s_x' \nabla^2_{xx} \ell(x^c, u^c) s_x$$

$$\text{subject to} \quad g(x^c) + \nabla g(x^c)' s_x \leq 0$$

and $u^c + s_u^c$ is the corresponding multiplier vector (see Boggs, Tolle, and Wang [1982]).

The SQP iteration for a nonlinear program having both equality and inequality constraints can now be described as follows:

[1] Given x^c, a current iterate, and Q_c, a current approximation of the Hessian of the Lagrangian, determine a Kuhn-Tucker point (s_x^c, u_q^c, v_q^c) of the quadratic program

$$\min_{s_x} \quad \nabla f(x^c)'s_x + (1/2)s_x'Q_c s_x$$

$$\text{subject to} \quad g_j(x^c) + \nabla g_j(x^c)'s_x \leq 0, \; j = 1, \ldots, m$$

$$h_k(x^c) + \nabla h_k(x^c)'s_x = 0, \; k = m+1, \ldots, m+p.$$

[2] Set

$$x^n = x^c + \alpha s_x^c, \; u^n = u_q^c, \; v^n = v_q^c$$

where α is a step-length parameter chosen so that an appropriate penalty function is decreased.

[3] Update Q_c so that Q_n is an approximation of

$$\nabla^2_{xx} \ell(x^n, u^n, v^n).$$

With the above iteration as a basis, the SQP method would be expected to have many of the properties of the well-known variable metric methods for unconstrained minimization, since in the absence of constraints, the SQP method reduces to a variable metric algorithm. For the constrained case, however, the Hessian (of the Lagrangian) need not be positive definite, thus the standard positive definite updates such as the BFGS or DFP (see Fletcher [1980] and Dennis and Schnabel [1983]) may not be appropriate. To date, local superlinear convergence has not been established for the use of positive definite updates except in the

case where the Hessian of the Lagrangian at the solution is positive definite. (See Han [1976] and Boggs, Tolle, and Wang [1982].)

There are nonpositive definite choices for the matrices Q_c that will lead to local superlinear convergence, as shown by Wilson [1963] (who uses the Hessian of the Lagrangian itself) and Han [1976]. But in these cases the solution of the quadratic program in step [1] is not straightforward; in fact, there may be multiple solutions. Nocedal and Overton [1983] have recently developed an updating scheme for equality constrained problems that may help to resolve this difficulty, but its application to general problems is not yet ensured.

In light of the absence of a feasible alternative, most optimizers implementing an SQP-type algorithm have opted for using positive definite updates of the BFGS or DFP type. The experimental results have been quite good (see Hock and Schittkowski [1981]) despite the lack of a solid theoretical underpinning. The approach of this paper is based on the same practical considerations, and hence a positive definite updating scheme will be used.

A vital part of an SQP algorithm, as for any algorithm for solving (1), is a provision for forcing convergence from a remote starting point. In the unconstrained case, this is accomplished by requiring a decrease in the objective function at each iterate. For constrained optimization, reduction of the objective function must be balanced

against satisfying the constraints. This balancing act is usually achieved by requiring a reduction in a "merit" or "penalty" function at each iteration. Han [1977] proposed that

$$(4) \quad p_r(x) = f(x) + r \left\{ \sum_{j=1}^{m} [g_j(x)]_+ + \sum_{k=m+1}^{m+p} |h_k(x)| \right\}$$

be used as a merit function, where r is larger than the absolute value of any of the multipliers associated with the solution to (1). Under reasonable conditions, he shows that any sequence $\{(x^k, u^k, v^k)\}$ generated using the SQP algorithm with a positive definite update and this merit function will converge to a Kuhn-Tucker point. The algorithm developed for this paper employs Han's merit function, thus guaranteeing global convergence under the conditions imposed by Han.

In spite of ensuring global convergence, this merit function is not entirely satisfactory. In particular, it does not guarantee that full steps, i.e., $\alpha = 1$, will be taken as x^k approaches x^*. This may restrict the convergence rate to be less than superlinear. Chamberlain, Lemarechal, Pedersen, and Powell [1979], Boggs and Tolle [1980, 1981], and Bertsekas [1980, 1981] have developed other merit functions that do allow for $\alpha = 1$ near a solution. However, these merit functions are computationally more complex than (4) and were not considered suitable for a large-scale algorithm.

General references on nonlinear programming methods that contain discussions of the SQP method and some of the other available methods include Fletcher [1981] and Gill, Murray, and Wright [1981].

I.2 STRUCTURE OF A SPARSE SQP ALGORITHM

The algorithm presented here is designed to solve large-scale nonlinear programming problems having sparse Lagrangian Hessians. Because it uses the SQP method, the algorithm is particularly well suited for solving problems having nonlinear constraints. The algorithm allows for any sparsity pattern in the Hessian of the Lagrangian, i.e., no particular sparsity pattern is assumed. Linear constraints and upper- and lower-bound constraints on the variables are handled explicitly by the algorithm. The number of constraints is theoretically unlimited; however, an active set strategy is used and the number of constraints active at any given time is constrained by the amount of memory available. Also, the algorithm is more efficient when the number of active constraints is small relative to the number of variables.

The following discussion describes the structure of the algorithm. The discussion focuses on the features of the algorithm that permit the SQP method to be applied to larger problems.

I.2.1 SPARSITY

The algorithm described here is designed to exploit the sparsity often found in large-scale problems. Such problems usually have sparse Lagrangian Hessians and sparse gradients. Handling sparse gradients is not difficult and most algorithms can be easily adapted to do so. However, the handling of a sparse Lagrangian Hessian is not so easy because, in general, matrix operations do not preserve sparsity. The Hessian of the Lagrangian, or an approximation of it, appears in most nonlinear optimization algorithms and will normally be used as the coefficient matrix of a system of equations that is used to compute a step direction as in the SQP algorithm. Exploiting the sparsity of the Hessian of the Lagrangian is important for two reasons. First, in large problems using the sparse structure often reduces the total amount of computation required. Second, storing a sparse matrix requires much less memory than storing a full matrix. Even if a computer has unlimited capacity for storing the matrix, such as is the case for virtual memory machines, manipulation of the full matrix may cause considerable paging of memory as different parts of the matrix are accessed. The resulting I/O time for swapping the different parts of the matrix between core and a high-speed storage device can be expensive.

We have chosen to store the representation of the Hessian of the Lagrangian as a lower triangular matrix under the assumption that the

approximation to the Hessian of the Lagrangian will be maintained as a positive definite matrix. Unfortunately, the Cholesky factor of a sparse symmetric and positive definite matrix need not be sparse. George and Liu [1981], however, describe an algorithm for permuting the rows and columns of a sparse, symmetric positive definite matrix that significantly reduces the fill-in that occurs in the lower triangular Cholesky factor. The authors also describe a storage scheme for the sparse matrix and code for solving systems of equations defined by the original matrix by performing forward and backward substitution on the triangular factor to obtain the solution.

Representing the Hessian approximation as the lower triangular factor has other computational advantages. The next section discusses the solution of the dual to the quadratic program defined by each iteration of the SQP algorithm. Generation of this dual problem is much simpler if the lower triangular factor is available. (See section II.3.4.)

I.2.2 SOLVING THE QUADRATIC PROGRAM

The quadratic program (QP) generated at each iteration of the SQP method has the form

$$(5) \quad \min_{s} \ (1/2)s'Qs + q's$$

$$\text{subject to} \quad As \leq a$$

$$Bs = b \ .$$

For a large-scale, sparse problem Q will be a large, sparse matrix. Standard methods for solving quadratic programs, such as the pivoting methods of large-scale linear programming methods, either do not take advantage of the sparsity structure or are too complicated for repeated use in a nonlinear programming code. The quadratic program generated by the SQP method may be infeasible. By solving the Wolfe dual (Wolfe [1961]) to (5) these problems can be avoided.

If (5) is feasible and Q is positive definite, then the Wolfe dual of (5) will be feasible and will have a nonempty solution set. The structure of this algorithm maintains a positive definite representation of Q. If the set $\{s : As \leq a, Bs = b\}$ is empty, the solution obtained from solving the Wolfe dual of (5) will be a "least infeasible" solution of (5). (See appendix B.) We solve a transformed version of the dual problem rather than the primal problem because it is a quadratic program having only nonnegativity constraints on variables that are the multipliers corresponding to the inequality constraints in (5). A projected preconditioned conjugate gradient algorithm for solving the transformed Wolfe dual problem is given in section I.3.

There are several advantages to solving the dual QP. If the number of active constraints is small relative to the number of variables in the nonlinear program, then the dual problem will be much smaller than problem (5). Moreover, since the solution to (5) is the step direction used by the SQP method, one would expect the step directions to change significantly from iteration to iteration, even when close to a solution of the nonlinear program. However, the solution to the dual problem can be shown to be a good approximation to the multiplier vector of the nonlinear program and should not change much from iteration to iteration. Consequently, a great efficiency is gained by using the estimated multiplier vector from the preceding iteration as the initial estimate in solving the dual.

Choi, Haug, Hou, and Sohoni [1983] report on the use of an algorithm developed by the Russian Pshenichny [1970] to solve optimal design problems. Pshenichny's algorithm is similar to the one described here in that he solves a sequence of quadratic programs by solving their duals. His sequence of quadratic programs is similar to Han's except that the matrix defining the quadratic program is always the identity, i.e., no second order information is used. Thus the algorithm is significantly different from that proposed herein.

I.2.3 STEP-LENGTH CONTROL

Having determined a step direction by obtaining the solution to (5), it is necessary to take a reasonable step in that direction. Since large problems are being solved, the method of controlling the step length must be relatively simple. We have chosen to use the penalty or merit function of Han [1977]

$$(6)\quad \phi(\alpha) = f(x + \alpha s) + r \left\{ \sum_{j=1}^{m} [g_j(x + \alpha s)]_+ + \sum_{k=m+1}^{m+p} |h_k(x + \alpha s)| \right\},$$

where x is the current estimate of the solution to the NLP and s is the step direction. The step-length parameter is α. The scalar, r, is chosen to be larger than the largest multiplier in absolute value. A step, αs, will be taken if an α can be found that produces an acceptable decrease from $\phi(0)$ to $\phi(\alpha)$. The details can be found in section I.4.

I.2.4 UPDATING PROCEDURES

The standard SQP method maintains an approximation to the Hessian of the Lagrangian which is updated after each iteration using one of the well-known matrix updating methods. One of the more common methods is the BFGS updating method (Dennis and More [1977]). The algorithm given here, however, maintains a sparse representation of the approximation to

the Hessian which the standard BFGS updating scheme does not do. Therefore, we have chosen a method of Goldfarb [1976] for updating the Cholesky factor of a positive definite matrix. Fill-in is ignored in applying the method. If fill-in were allowed, the method would produce the standard BFGS update for the Cholesky factor. Details can be found in section I.5.

I.2.5 A BASIC ITERATION OF THE ALGORITHM

The following is a description of a basic iteration of the algorithm developed here. The algorithm uses an active set strategy. Equality constraints are always active and inequality constraints are active at the current iterate if they are infeasible or nearly so at that point. Upper- and lower-bound constraints are treated as general inequality constraints in the description of a basic iteration. The actual implementation, as described in Part II, explicitly handles upper- and lower-bound constraints.

[1] Solve the transformed QP for the multiplier vector. Let (x^c, u^c, v^c) and L_c be the current estimate of the Kuhn-Tucker vector and the approximate lower triangular factor of the Lagrangian Hessian, respectively. The QP generated by the standard SQP method is

$$(7)\quad \min_{s \in R^n} \quad \nabla f(x^c)'s + (1/2)s'L_cL_c's$$

$$\text{subject to:} \quad \nabla g(x^c)'s + g(x^c) \leq 0$$

$$\nabla h(x^c)'s + h(x^c) = 0 \ .$$

Let $\bar{g}$ be the vector of active inequality constraints -- including active upper- and lower-bound constraints. The solution to (7) and the associated multiplier vectors are obtained by transforming (7) into

$$(8)\quad \min_{u \in R^{m'},\ v \in R^p} \quad (1/2)(u', v')\ K\ (u', v')' + (u', v')\ k$$

$$\text{subject to} \quad u \geq 0$$

where

$$K = (\nabla\bar{g}(x^c),\ \nabla h(x^c))'L_c^{-1'}L_c^{-1}(\nabla\bar{g}(x^c),\ \nabla h(x^c))$$

and

$$k = (\nabla\bar{g}(x^c),\ \nabla h(x^c))'L_c^{-1'}L_c^{-1}\nabla f(x^c) - (\bar{g}(x^c)',\ h(x^c)')',$$

and m' is the number of active inequality constraints, including active upper- and lower-bound constraints. (The number of active inequality constraints, m', may change from iteration to iteration.) Let $p' = m' + p$. Then K is the $p' \times p'$ matrix defined by $K = M'M$, where

$$M = L_c^{-1}(\nabla\bar{g}(x^c),\ \nabla h(x^c)).$$

The matrix M can be computed by p' forward substitutions using the lower triangular matrix L_c. Once M is formed, another forward substitution produces $q = L_c^{-1}\nabla f(x^c)$ so that $k = M'q - (\bar{g}(x^c)',\ h(x^c)')'$.
A projected preconditioned conjugate gradient algorithm is employed to solve (8).

Let $(u^{n'},\ v^{n'})'$ be the solution to (8).

[2] Solve for the step direction. Let s^c be the solution to

$$L_cL_c's = -\{\nabla f(x^c) + \nabla g(x^c)u^n + \nabla h(x^c)v^n\}.$$

(If inequality constraint g_j is considered inactive, then u_j is set to 0.) If QP (7) is feasible, then s^c is its solution. If QP (7) is not feasible, then an s^c is obtained which is a point of minimal infeasibility. Note that only one forward substitution and one backward substitution are required to solve for s.

[3] Compute the step-length. Let $r > \max_{i,\ j}\ \{|u_i^n|,\ |v_j^n|\}$

and

$$\phi(\alpha) = f(x^c + \alpha s^c) + r\{ \sum_{i=1}^{m} [g_i(x^c + \alpha s^c)]_+$$

$$+ \sum_{j=1}^{p} |h_j(x^c + \alpha s^c)|\}.$$

Choose α^c such that $0 < \alpha^c \leq 1.0$ and $\phi(\alpha^c)$ is sufficiently smaller than $\phi(0)$. Set $x^n = x^c + \alpha^c s^c$.

[4] Check for termination. Compute the gradient of the Lagrangian function at (x^n, u^n, v^n), $\nabla_x \ell(x^n, u^n, v^n)$, and terminate successfully if $|\nabla_x \ell(x^n, u^n, v^n)|$ is small relative to the objective function value.

[5] Update the triangular factorization. Update L_c but maintain the sparsity structure. Use the BFGS updating procedure for the Cholesky factor of a positive definite Hessian developed by Goldfarb [1976] with a modification that preserves the sparsity pattern in L_c.

[6] Go to [1] for the next iteration.

The following sections of Part I discuss these steps in more detail. Part II describes the implementation of the algorithm and provides many of the details not given here, such as what to do if no acceptable step is possible in the direction s^c.

I.3 THE QUADRATIC PROBLEM

Transforming the general quadratic program defined in (7) into the quadratic program having only nonnegativity constraints on some of the variables (8) has two advantages. First, if the number of active constraints in the original problem is small relative to the number of variables, then the transformed QP will be much smaller than the original and will have simple constraints. Second, solving the transformed problem with a conjugate gradient method has proved to be very efficient (see Part II). This is especially true when near a solution as the initial estimate of the solution to the transformed problem will be close to the multipliers for (1) and will not change much from iteration to iteration. The conjugate gradient algorithm will therefore have to do very little work to refine the estimates on each iteration. In constrast, the solution to the general quadratic program (7), being the step direction, will change significantly from iteration to iteration. Hence, using the step direction from the preceding iteration as the initial estimate of the solution will not improve computational efficiency. If the original QP (7) is infeasible, the transformed problem is still feasible but unbounded, though it can be made strictly convex by a simple adjustment. Infeasibility of the

original QP should occur only when far from a solution to (1), and it will be shown that the step direction computed using the approximate multipliers obtained from the adjusted version of (8) will allow the algorithm to continue making progress.

Problem (7) could be solved using one of the pivoting algorithms (see Dantzig [1963] and Beale [1967]). These algorithms, however, are not particularly useful for solving large, sparse problems because they destroy sparsity. Thus, they are not considered useful for solving (7) in the context addressed here. They were also not considered for solving the transformed problem (8) even though these problems should be smaller than (7) and, possibly, denser. The reason is that the pivoting methods cannot be used to refine an estimate of a solution that is already close to the desired result. Pivoting methods do not start with an estimate of the solution so, unlike the CG methods, they do not exhibit a decrease in computation when a good estimate of the solution is already available.

I.3.1 THE WOLFE DUAL

Suppose Q is a positive definite n x n matrix, A and B are, respectively, m x n and p x n matrices, and q, a, and b are fixed vectors of appropriate dimension. The general, strictly convex quadratic problem has the form

$$(9) \quad \min_{s \in R^n} \quad (1/2)s'Qs + q's$$

$$\text{subject to} \quad As \leq a$$

$$Bs = b.$$

The Wolfe dual to this problem (Wolfe [1961]) can be written

$$(10) \quad \max_{(s, u, v) \in R^{n+m+p}} \quad (1/2)s'Qs + q's + (As - a)'u + (Bs - b)'v$$

$$\text{subject to} \quad Qs + q + A'u + B'v = 0$$

$$u \geq 0.$$

In the case where $Q \equiv 0$, problems (9) and (10) become the standard dual pair of linear programs.

Since Q is positive definite, the equality constraint in (10) can be solved for s and then s can be replaced in the objective function. Thus, letting $w = (u', v')' \in R^{m+p}$, (10) can be written as

$$(11)\quad \min_{w \in R^{m+p}} \quad (1/2)w'Kw + k'w$$

$$\text{subject to} \quad u \geq 0,$$

where $K = VQ^{-1}V'$, $V = (A', B')'$, and $k = VQ^{-1}q + v$. Here $v = (a', b')'$. Note that K will be positive definite if and only if V has full row rank, else it will be positive semi-definite. In the former case the following theorem is well known.

Theorem 1: Suppose $V = (A', B')'$ has full row rank, then both (9) and (11) have unique solutions, say s^* and $w^* = (u^{*\prime}, v^{*\prime})'$, w^* is the multiplier vector for (9), and

$$(12)\quad s^* = -Q^{-1}[A'u^* + B'v^* + q].$$

Sometimes, however, there are enough inequality constraints so that $m + p > n$. Then V cannot have full row rank and the above result does not apply. However, if (9) is feasible, we have the following result. (See Wolfe [1961].)

Theorem 2: Suppose (9) is feasible. Then problem (9) has a unique solution s^* and problem (11) has a nonempty solution set W^*. Moreover, for any $w^* = (u^{*\prime}, v^{*\prime})' \in W^*$, equation (12) holds. If (9) is

infeasible, then problem (11) is unbounded and has no solution.

In the application of problem (9) in the algorithm given here, it is possible that the quadratic problem may be infeasible. In this case, the following result will be applicable.

Consider the perturbed version of (11):

$$(13)\quad \min_{w \in R^{m+p}} \quad (1/2)w'(K + \varepsilon I)w + k'w$$

$$\text{subject to} \quad u \geq 0$$

where ε is a small positive number. Since $K + \varepsilon I$ is a positive definite matrix, problem (13) has a unique solution $w^{\varepsilon} = (u^{\varepsilon}, v^{\varepsilon})$. We denote by

$$(14)\quad s^{\varepsilon} = -Q^{-1}[A'u^{\varepsilon} + B'v^{\varepsilon} + q].$$

For a given s vector we measure its infeasibility in the original quadratic problem (9) by

$$e(s) = \{|[As - a]_+|_2^2 + |Bs - b|_2^2\}^{1/2}.$$

Then $e(s) = 0$ if and only if s is feasible for (9). The set of least infeasible points is denoted by

$$Z = \{s: e(s) \leq e(t) \text{ for all } t \in R^n\}.$$

Clearly, Z is a convex, closed, nonempty subset of R^n. If (9) is feasible, it is exactly the feasible set.

Theorem 3: Let $\{w^\varepsilon\}$ be the family of solutions to (13) for positive values of ε, and for each ε let s^ε be given by (14). Then

$$\lim_{\varepsilon \to 0^+} s^\varepsilon = \hat{s}$$

where $\hat{s}$ is a solution of the problem

$$(15) \quad \min_{s \in Z} \quad (1/2)s'Qs + q's.$$

Pf: See Appendix B.

In the algorithm presented for solving problem (1), Q is the updated approximation of the Hessian matrix, which is positive definite. A is taken to be the gradients of the active inequality constraints and B the gradients of the equality constraints at the current iterate x^c. By active constraint, we include all of the equality constraints and any inequality constraints for which $g_j(x^c) > -\eta$, where $\eta > 0$ is a prescribed tolerance. Assuming feasibility and nondegeneracy for the original problem, the quadratic programs to be solved will likely have less than full row rank only when

the approximation is far from feasibility. In this case, the perturbed dual problems will be solved with small ε. Theorem 3 provides justification for this procedure in that it ensures that the solution s^{ε} will be a step toward minimum infeasibility.

I.3.2 THE CONJUGATE GRADIENT ALGORITHM

Before considering the application of the conjugate gradient method to the minimization of a quadratic function subject to nonnegativity constraints, we should review some of its properties when applied to the unconstrained minimization of a quadratic function. The conjugate gradient method for solving

$$(16) \quad \min_{w \in R^n} \quad F(w) = (1/2)w'Kw + k'w$$

is as follows:

[0] Starting at any $w^0 \in R^n$ set $\ell = 0$, and define $p^0 = -\nabla F(w^0) = -Kw^0 - k$.

[1] Set $w^{\ell+1} = w^{\ell} + \alpha_{\ell} p^{\ell}$,

where $\alpha_{\ell} = - \dfrac{\nabla F(w^{\ell})' \, p^{\ell}}{p^{\ell\prime} \, K \, p^{\ell}}$.

[2] If $\nabla F(w^{\ell+1}) = 0$, set $w^* = w^{\ell+1}$ and terminate with w^* as the solution to (16). Otherwise, go to [3].

[3] Set $p^{\ell+1} = -\nabla F(w^{\ell+1}) + \beta_\ell\, p^\ell$,

where $\beta_\ell = \dfrac{\nabla F(w^{\ell+1})'\, K\, p^\ell}{p^{\ell\prime}\, K\, p^\ell}$.

[4] Set $\ell = \ell + 1$. Go to [1].

In exact arithmetic, the algorithm terminates in at most n iterations for positive definite K. The conjugate gradient algorithm converges monotonically to w^* in that if we define

$$E(w) = (w - w^*)'\, K\, (w - w^*)$$

then it is easy to show that

$$E(w^{\ell+1}) = E(w^\ell) - \frac{[\nabla F(w^\ell)'\, p^\ell]^2}{p^{\ell\prime}\, K\, p^\ell} > E(w^\ell).$$

Thus, in the metric defined by the positive definite matrix K, the conjugate gradient estimates get closer to the solution on each iteration. Likewise, it is easy to show that

$$F(w^{\ell+1}) - F(w^\ell) = -\frac{1}{2}\frac{[\nabla F(w^\ell)'\, p^\ell]^2}{p^{\ell\prime}\, K\, p^\ell},$$

showing that the objective function is decreased on each iteration. It should be noted that solving (16) for positive definite K is equivalent to solving $Kw + k = 0$ for w.

If we let $r^{\ell} = Kw^{\ell} + q$, we can think of the CG algorithm as either trying to find a zero of the gradient or trying to make the residual associated with the linear equation, r, equal to zero. Unlike most methods for solving linear systems of equations, the CG method does not alter the matrix K and involves only matrix-vector multiplications.

The finite termination property and the monotone decrease in the distance between w^{ℓ} and w^* as defined by the matrix norm are also achieved by a modification to the conjugate gradient algorithm (see Polyak [1969]) which minimizes a quadratic function subject to nonnegativity constraints on the variables.

Let $y = Kw + k$. Then w^* solves (16) if and only if

(17) $y_i^* > 0$ if $w_i^* = 0$ and

(18) $y_i^* = 0$ if $w_i^* > 0$.

These are the Kuhn-Tucker conditions for the solution to (16). Another way of stating these conditions is to say that w^* solves (16) if and only if

(19) $w^{*\prime}y^{*} = 0$ (complementarity condition) and

(20) $w^{*}, y^{*} \geq 0$ (nonnegativity condition).

Polyak's algorithm, which is the basis of the algorithm developed by O'Leary [1981], maintains nonnegativity of the vector iterates w^{ℓ} while iterating toward satisfying the remaining conditions in (19) and (20). Polyak's algorithm terminates in a finite number of iterations (O'Leary [1981]).

I.3.3 A PROJECTED PRECONDITIONED CONJUGATE GRADIENT (PPCG) ALGORITHM

O'Leary [1981] describes a modification to Polyak's algorithm that preconditions the CG step to improve the convergence rate of the algorithm. She also proves that her algorithm converges in a finite number of iterations. O'Leary's algorithm has been modified for the work described in this application. The standard CG step is projected onto the feasible region as opposed to O'Leary's method of truncating the step at the boundary of the feasible region. Taking projected steps has the advantage of allowing more than a single variable to become inactive on an iteration, while truncation will permit only one variable to become inactive on a single iteration.

A preconditioned conjugate gradient algorithm for (11) uses $M^{-1}\nabla F(w^{\ell})$ rather than $\nabla F(w^{\ell})$ to define $p^{\ell+1}$ in step [3] of the CG algorithm, where M^{-1} is some approximation to K^{-1}. One obvious choice for M^{-1} is the inverse of the diagonal of K. Another choice, and the one actually employed here, is one pass of the symmetric successive over-relaxation (SSOR) method as applied to the system of equations defined by $Kw + k = 0$. The following is a description of the PPCG algorithm.

In our description of the PPCG algorithm, N will be the set of indices of components of w that are constrained to be nonnegative and $\bar{N}$ will be the set of remaining indices. A vector w will be a feasible solution to the quadratic program if $w_i \geq 0$ for $i \in N$. At any given time, a variable will be considered active or inactive. Only constrained variables can be inactive. An inactive variable will always be at its bound, i.e., equal to zero. A variable, however, may be at its bound but not considered inactive.

The PPCG Algorithm

[0] (Initialization) Choose w^0 such that $w_i^0 \geq 0$ for $i \in N$ and set $\ell = 0$. Set $I = N$.

[1] (Outer iteration) Set $\ell = \ell + 1$, $y^{\ell} = Kw^{\ell} + k$, and $I_{\ell-1} = I$. Define $I_{\ell} = \{i \in N: w_i^{\ell} = 0 \text{ and } y_i > 0\}$. If $I_{\ell} = I_{\ell-1}$ and $|y_i| < \varepsilon$ for $i \notin I_{\ell}$, then terminate; otherwise, set $I = I_{\ell}$. (Note: ε is the tolerance on the norm of the residuals for termination.)

[2] (Inner iteration) The inner iteration only manipulates active variables. During the inner iteration, variables that are active may become inactive, but no inactive variables become active. Inactive variables can become active only during execution of step [1]. Let J be the set of indices of active variables, then any variable index belongs either to I or J. The matrix system is partitioned as follows:

$$w^{\ell} \rightarrow \begin{vmatrix} w_I^{\ell} \\ w_J^{\ell} \end{vmatrix}, \quad k \rightarrow \begin{vmatrix} k_I \\ k_J \end{vmatrix}, \quad K \rightarrow \begin{vmatrix} K_{II} & K'_{JI} \\ K_{JI} & K_{JJ} \end{vmatrix}$$

Initialize to solve the equation

$$(21) \quad K_{JJ}\, w_J = -k_J - K_{JI}\, w_I.$$

Set $z^0 = w_J^{\ell}$ and $r^0 = -k_J - K_{JI}\, w_I^{\ell} - K_{JJ}\, z^0$. Go to step [4].

[3] (Restart inner iteration) If a projected step was taken, the variables that have been set to their bounds must be checked to see whether they should become inactive variables. Also, the residual

vector must be recomputed since the CG formula for updating the residual is not valid for a projected step. Set $z^0 = w_J^\ell$ and $r^0 = -k_J - K_{JI}\, w_I^\ell - K_{JJ}\, z^0$. For each $i \in J \cap N$ do the following: if $z_i^0 = 0$ and $r_i^0 < 0$, then add i to I. Repartition as necessary. If there are no active variables, then restart the outer iteration (go to [1]).

[4] (Calculate new iterate and residual) Set

$$a_{CG} = \frac{r^{0\prime}\, r^0}{r^{0\prime}\, K_{JJ}\, r^0} .$$

Set $z^1 = z^0 + a_{CG}\, r^0$. For each $i \in J \cap N$, do the following: set $z_i^1 = [z_i^1]_+$ and set a projection flag if this is a projected step. (Note: O'Leary's algorithm allows only one variable per iteration to become inactive, whereas this one may set more than one to the inactive state on a single iteration.)

If this is a projected step, set $w_J^\ell = z^1$ and go to [3]. Otherwise, set $r^1 = r^0 - a_{CG}\, K_{JJ}\, r^0$.

[5] If $| r^1 | < \varepsilon$, set $w_J^\ell = z^1$ and restart the outer iteration (go to [1]).

[6] (Initialize preconditioned iteration) Choose M as a preconditioning matrix for K_{JJ}, set $q = 1$, and let $p' = M^{-1}r'$.

[7] (Calculate new iterate and residual) Set

$$a_{CG} = \frac{r^{q'} p^q}{p^{q'} K_{JJ} p^q} = \frac{r^{q'} M^{-1} r^q}{p^{q'} K_{JJ} p^q},$$

$$z^{q+1} = z^q + a_{CG} p^q$$

For each $i \in J \cap N$, do the following: set $z_i^{q+1} = [z_i^{q+1}]_+$ and set the projection flag if this is a projection step. If this is a projected step, set $w_J^{\ell} = z^{q+1}$ and go to [3]. Otherwise, set

$$r^{q+1} = r^q - a_{CG} K_{JJ} p^q.$$

[8] If $\left| r^{q+1} \right| < \varepsilon$, set $w_J^{\ell} = z^{q+1}$ and restart the outer iteration (go to [1]).

[9] (Calculate new search direction) Set

$$b_q = -\frac{p^q,\ K_{JJ}\ M^{-1}\ r^{q+1}}{p^q,\ K_{JJ}\ p^q} = \frac{r^{(q+1)},\ M^{-1}\ r^{q+1}}{r^q,\ M^{-1}\ r^q},$$

$$p_q = M^{-1}r^{q+1} + b_q p^q$$

$$q = q + 1.$$

Go to [7] for the next preconditioned step.

O'Leary's algorithm does not take projected steps; instead, a step is truncated at the first boundary it encounters. If M is set to the identity in O'Leary's algorithm, then the algorithm is identical to Polyak's. As long as the preconditioning matrix, M, is positive definite, O'Leary's algorithm converges after a finite number of iterations. The projected version of this algorithm has performed well on the problems used for testing the algorithm developed in this paper; however, a thorough investigation of its properties remains to be done.

Possible choices for M include the diagonal of K_{JJ}. M is clearly positive definite in this case if K is positive definite, so for O'Leary's algorithm the finite convergence property will hold. Another choice, which O'Leary investigated, is to define $M^{-1}r$ as follows. Let

$$M^{-1}r = \bar{z} - z^i$$

where $\bar{z}$ is the vector obtained by applying one iteration of the symmetric successive over-relaxation (SSOR) algorithm to system (21) with z^i as the current estimate of w_J. In O'Leary's version of this application of the SSOR method, variables are truncated during the forward and backward passes. The M corresponding to this process is not necessarily positive definite, so the properties of the algorithm are unknown. However, O'Leary reported good results with this preconditioning method. For the projected algorithm described here, the variables are not truncated during the forward and backward passes of the SSOR method. Consequently, the preconditioning matrix M defined by this process is positive definite (see Hageman and Young [1981]) and should make the investigation of the properties of the projected CG algorithm easier. The following is a description of the forward and backward passes of one iteration of the SSOR method.

SSOR Algorithm

[1] Let $F_J = k_J - K_{JI}w_I$.

[2] For $j = 1, 2, \ldots, s$, where s is the order of K_{JJ}, set

$$\bar{z}^f_j = z^i_j + \omega \left(F_j - \sum_{\ell=1}^{j-1} K_{j\ell}\bar{z}^f_\ell - \sum_{\ell=j}^{s} K_{j\ell}z^i_j\right)/K_{jj} ,$$

and $\omega \in (1, 2)$ is the relaxation parameter.

[3] For $j = s, s - 1, \ldots, 1$ let

$$\bar{z}_j = \bar{z}^f_j + \omega \left(F_j - \sum_{\ell=1}^{j} K_{j\ell}\bar{z}^f_\ell - \sum_{\ell=j+1}^{s} K_{j\ell}\bar{z}_\ell\right)/K_{jj} .$$

The SSOR method can be used to solve systems of equations defined by a positive definite matrix. As a preconditioning step for a CG algorithm, a single SSOR iteration should provide a refined estimate for the next step direction.

I.4 MERIT FUNCTION FOR STEP-LENGTH CONTROL

We use Han's merit function for step-length control. Suppose the current iterate is x^c and the step vector is s^c, then let $\phi(\alpha) = p_r(x^c + \alpha s^c)$, where $p_r(.)$ is given by (4) and α, $0 < \alpha \leq 1$, is the step-length parameter. We set $x^n = x^c + \alpha s^c$ if $\phi(\alpha) \leq \phi(0) + \sigma\, \alpha\phi'(0)$, where $0 < \sigma \leq 1$. The derivative, $\phi'(0)$, is the right-hand derivative of ϕ at 0 and is computed as follows.

Define

$$G_j = \begin{cases} 0: \; g_j(x^c) < 0 \text{ or } [g_j(x^c) = 0 \\ \qquad \text{and } \nabla g_j(x^c)'s^c < 0] \\ \nabla g_j(x^c)'s^c: \text{ otherwise} \end{cases}$$

for inequality constraints, and

$$H_k = \begin{cases} -\nabla h_k(x^c)'s^c: & h_k(x^c) < 0, \\ \nabla h_k(x^c)'s^c: & h_k(x^c) > 0, \\ \left|\nabla h_k(x^c)'s^c\right|: & h_k(x^c) = 0, \end{cases}$$

for equality constraints. Then $\phi'(0)$ is given by

$$(22) \quad \phi'(0) = \nabla f(x^c)'s + r \left\{ \sum_{j=1}^{m} G_j + \sum_{k=m+1}^{m+p} H_k \right\},$$

where r is larger than the absolute value of any of the multipliers.

The choice of σ determines the strictness of the test. Normally, σ is set to 0.1. Note that in general, $\phi(.)$ is a continuous, but not necessarily smooth, function of α. It is still the case, however, that for some $\alpha \in (0, \hat{\alpha})$ with $0 < \hat{\alpha} \leq 1$ the test can be passed if s is really a descent direction for $p_r(.)$.

Han's [1977] algorithm differs from the one developed here in that the new iterate, x^{k+1}, is given by

$$x^{k+1} = x^k + \alpha^k s^k$$

for any α^k in $[0, \zeta]$ satisfying

$$p_r(x^{k+1}) \leq \min_{0 \leq \alpha \leq \zeta} p_r(x^k + \alpha s^k) + \varepsilon^k$$

with $\{\varepsilon^k\}$ a sequence of numbers satisfying

$$\sum_{k=0}^{\infty} \varepsilon^k < \infty$$

and ζ is some positive number. Han shows that his algorithm, with a proper choice of r for the merit function (4), is globally convergent under the following conditions:

(i) f, g_j, $j = 1, \ldots, m$, are continuously differentiable;

(ii) f is strictly convex and bounded below;

(iii) the constraint functions are convex;

(iv) the set $X = \{x: g(x) \leq 0\}$ is compact and $X^o = \{x: g(x) < 0\} \neq \emptyset$;

(v) there exist positive numbers λ_1 and λ_2 such that for each k and for any $x \in R^n$,

$$\lambda_1 x'x \leq x'Q_k x \leq \lambda_2 x'x.$$

This result can be extended to include equality constraints. In the same paper, Han proves a weaker global convergence property requiring only conditions (i) and (v) and that each quadratic program generated by the algorithm have a Kuhn-Tucker point with a Lagrange multiplier vector bounded by r in ∞-norm. Note that these global convergence results require only that the Hessian approximation matrices, Q_k, be positive definite, with their eigenvalues bounded above and below. The step-

length control parameter used in the algorithm developed here uses the same merit function as Han, but we do not require the nearly exact minimization over α as Han does. Instead, we require the step to achieve a rate of descent compatible with the local behavior of the problem functions. This approach is similar to the Goldstein-Armijo principle (see Fletcher [1980]) and has performed well in testing (see Part II).

I.5 SPARSE UPDATING PROCEDURE

Variable metric algorithms for unconstrained minimization update approximations to the Hessian in such a way that the quasi-Newton condition is satisfied. Let f be the objective function of an unconstrained minimization problem and let Q_n be the updated approximation to the Hessian. Then the updating procedure used to obtain Q_n satisfies the quasi-Newton condition if

$$Q_n(x^n - x^c) = \nabla f(x^n) - \nabla f(x^c).$$

Updates satisfying the quasi-Newton condition, such as the BFGS update, have many desirable properties, including superlinear convergence (see Fletcher [1980]). If the Hessian of f is sparse, it is advantageous if the updating procedure maintains the sparsity pattern. Shanno [1980] has shown, however, that it is not, in general, possible to have an updating scheme that satisfies the quasi-Newton condition, maintains a given sparsity pattern, and preserves positive definiteness.

The same problem arises for the constrained problem where the quasi-Newton condition is

$$(23) \quad Q_n(x^n - x^c) = \nabla_x \ell(x^n, u^n, v^n) - \nabla_x \ell(x^c, u^n, v^n).$$

As discussed earlier, the solution of the quadratic subproblems requires that the updating scheme be positive definite. Moreover, for solving large problems, maintaining the sparsity pattern of the Lagrangian Hessian is essential. Therefore, the requirement that the update satisfy (23) has been dropped in favor of maintaining the sparsity pattern. The update is forced to have the desired sparsity pattern by "zeroing out" the appropriate elements in the lower triangular factor of the update. The effect of this decision on the local convergence rate is unknown even in the unconstrained case. Test results, however, have been encouraging. Thapa [1983] reports favorable results for this type of procedure applied to the BFGS update for unconstrained optimization as long as the updated factor remains positive definite.

Since the algorithm developed here maintains a sparse lower triangular approximation to the Cholesky factor of the Hessian, a procedure for updating the lower triangular approximation is used. The procedure is a modification of Goldfarb's [1976] BFGS procedure for updating the Cholesky factor of a positive definite Hessian approximation. The procedure is simple to implement. Other methods for updating a sparse Hessian approximation have been developed by

Shanno [1980] and Toint [1977]. These methods were not used because of their complexity and because they are not directly applicable to updating a lower triangular factor.

Goldfarb Updating Procedure

Let $s^n = x^n - x^c$ and let L_c be the current approximation to the lower triangular factor of the Hessian of the Lagrangian, i.e., let $Q_c = L_c L_c'$. The BFGS update for Q_c is given by

$$Q_n = Q_c + \frac{yy'}{s^n{}'y} - \frac{Q_c s^n s^n{}' Q_c}{s^n{}' Q_c s^n},$$

where $y = \nabla_x \ell(x^n, u^n, v^n) - \nabla_x \ell(x^c, u^n, v^n)$. Let $p = s^n$ and

$$q = y/[(s^n{}'y)(s^n{}'Q_c s^n)]^{1/2} - Q_c s^n / s^n{}' Q_c s^n.$$

Define z and w by $L_c z = q$ and $w = L_c{}' p$. Then

$$Q_n = (I + qp')L_c L_c'(I + pq')$$

$$= L_c(I + zw')(I + wz')L_c'.$$

We wish to find $\tilde{L}$ such that $(I + zw') = \tilde{L}Q'$, with $\tilde{L}$ lower triangular

and Q orthogonal. Then

$$Q_n = L_c\tilde{L}\, Q'Q\, \tilde{L}'L_c' = L_c\tilde{L}(L_c\tilde{L})'$$

so that $L_n = L_c\tilde{L}$ is the new approximation to the Cholesky factorization of the Lagrangian. L_n will not necessarily have the same sparsity structure as L_c; however, it is simple to ignore fill-in in L_n. If $w'z = -1$ so that $I + zw'$ is singular, then we cannot update. However, $w'z = -1$ occurs only if $s'y = 0$. If $s'y < 0$, a modification suggested by Powell [1978] is used that maintains the positive definiteness of the update (see Part II).

The $\tilde{L}$ sought is given by

$$\tilde{L}_{ij} = \begin{cases} \rho_i: & i = j \\ \beta_j w_i + \gamma_j z_i: & i > j \\ 0: & \text{otherwise} \end{cases}$$

for $i, j = 1, \ldots, n$. The following two recurrences generate the vectors β, γ, ρ.

Recurrence 1:

1. Set $\bar{\beta}_n = 1/w_n$.

2. For $j = n-1, n-2, \ldots, 1$ set

$$r_j = \bar{\beta}_{j+1} \cdot w_j$$

$$s_j = (r_j^2 + 1)^{-1/2}$$

$$c_j = r_j \, s_j$$

$$\bar{\beta}_j = s_j \, \bar{\beta}_{j+1}$$

$$\bar{\beta}_{j+1} = c_j \, \bar{\beta}_{j+1}.$$

Recurrence 2:

1. Set $\bar{\gamma}_1 = 1/\bar{\beta}_1$, $\bar{\rho}_1 = \bar{\beta}_1 \, w_1 + \bar{\gamma}_1 \, z_1$

2. For $j = 1, 2, \ldots, n-1$ set

$$\rho_j = (\bar{\rho}_j^2 + s_j^2)^{1/2}$$

$$\bar{c}_j = \bar{\rho}_j / \rho_j$$

$$\bar{s}_j = s_j / \rho_j$$

$$\beta_j = \bar{c}_j \, \bar{\beta}_j - \bar{s}_j \, \bar{\beta}_{j+1}$$

$$\gamma_j = \bar{c}_j \bar{\gamma}_j$$

$$\bar{\gamma}_{j+1} = \bar{s}_j \bar{\gamma}_j$$

$$\bar{\beta}_{j+1} = \bar{s}_j \bar{\beta}_j + \bar{c}_j \bar{\beta}_{j+1}$$

$$\bar{\beta}_{j+1} = \bar{\beta}_{j+1} w_{j+1} + \bar{\gamma}_{j+1} z_{j+1}$$

3. Set $\rho_n = \bar{\rho}_n$.

Since $L_n = L_c \tilde{L}$, the j-th column of L_n is given by

$$(L_n)_{.j} = \sum_{k=j}^{n} \tilde{L}_{jk} (L_c)_{.k}$$

$$= \rho_j (L_c)_{.j} + \sum_{k=j+1}^{n} (\beta_j w_k + \gamma_j z_k)(L_c)_{.k}, \quad j = 1, \ldots, n.$$

Thus, L_c can be updated without explicitly forming $\tilde{L}$. Only the vectors ρ, β, γ, w, z need be stored. Any fill-in is ignored.

PART II. IMPLEMENTATION AND TEST RESULTS

II.1 INTRODUCTION

This part describes the computer implementation of the algorithm and gives the results obtained by applying the algorithm to several test problems. An assessment of the capabilities of the algorithm based on the test results is also given.

The computer implementation of the algorithm is referred to as OPCON. The discussion here focuses on those issues that are of particular importance in implementing the algorithm on a computer.

The notation used in this part will be the same as in Part I, with the exception that references to equations in Part I will be preceded by "I," e.g., (I.16) will refer to equation 16 in Part I.

II.2 PRELIMINARIES

Before discussing the computer implementation, it would be helpful to describe how the implementation handles function evaluation, gradient storage, and gradient estimation.

II.2.1 FUNCTION EVALUATION SUBROUTINE

The user must supply a subroutine that computes the value of the objective function and all of the nonlinear constraint functions at a point passed to the subroutine. Linear constraints are handled by providing the coefficients as part of the gradient data. Each call to the function evaluation routine results in the evaluation of the objective function and all of the nonlinear constraints.

A smoother version of the OPCON algorithm would allow a separate call to evaluate the objective function and another call to evaluate the nonlinear constraints. This feature would be useful in problems having all variables represented in the objective but having sparse constraint functions. The sparse finite differencing gradient procedures of Coleman and More [1982], discussed in the following section, would

provide more savings because the method could be applied to just the set of constraint functions. A single non-sparse function in the set of functions handled by the Coleman and More method will result in no savings in function evaluations. For example, if two functions of variables of length two have gradients that are structurally zero in opposite components, then one call to a function evaluation routine for both functions is enough to obtain forward difference estimates for the two nonzero components. If one of the functions has all nonzero gradient elements, then two calls will be required. The Coleman and More procedure exploits these relationships. If one function in a set is not sparse, then the number of calls to the function evaluation subroutine will be nearly the number of variables.

II.2.2 FINITE DIFFERENCING AND SPARSE GRADIENTS

The nonzero gradient elements for all nonlinear functions are estimated from either forward or central differencing. Since all functions are evaluated by each call to the function evaluation subroutine, it is worthwhile to reduce the number of calls of the subroutine required to estimate all nonzero gradient elements. If the gradients of the nonlinear functions were not sparse, then n calls of the function evaluation subroutine would be required, where n is the number of variables in the problem. If, however, the gradients are sparse, it is possible to significantly reduce the number of calls as

Coleman and More [1982] have shown. Their algorithm has been employed in the development of OPCON.

A single value for the finite differencing interval that is used for all variables is input to the algorithm. However, each restart of the algorithm (restarts occur after failures to take a step) causes a finite difference interval to be computed for each variable. This is accomplished by a call for each variable to the subroutine, FDCALC, developed by Gill, Murray, Saunders, and Wright [1981] for determining good finite differencing intervals. Their procedure balances truncation error against the noise induced by machine evaluation of the function. Since the step direction chosen at each iteration is the solution to a system of equations having an estimate of the Lagrangian gradient as the right-hand-side, choosing the finite difference interval for each variable to make the finite difference estimate of the Lagrangian gradient reasonably accurate is appropriate. Thus, the Lagrangian function using the current estimate of the multiplier vector is the function passed to these subroutines. This is an appropriate choice since it is the estimated gradient of the Lagrangian which is used to determine the step direction.

As mentioned previously, it is important to reduce the number of calls to the function evaluation subroutine in obtaining the finite difference estimates for the sparse gradients. The sparse gradient structure of the objective function and nonlinear constraint functions

is processed to obtain n_G groups of variables, such that the variables in each group can be varied together to compute some of the finite difference elements of the Jacobian matrix corresponding to the gradients of these functions. Only n_G calls to the function evaluation subroutine are required to compute all the structurally nonzero entries in the forward difference estimate of the sparse Jacobian matrix. If n_G is significantly smaller than n, then there is a considerable savings in the number of computations required to compute the estimate. Since central differencing is used whenever there is an indication that forward differencing may not be sufficiently accurate, the savings can be even more pronounced. (See Gill, Murray, Saunders, and Wright [1981], Stewart [1967], and section II.3.3 for further discussion of when to switch to central differencing.)

Linear constraints are handled separately from nonlinear constraints. The coefficients are stored in a sparse format and used to compute function values or gradients as needed. These arrays are passed to OPCON. Each linear or nonlinear constraint is set to be either an inequality (<) or equality constraint. The user also sets the right-hand-side (RHS) value of each constraint. An initial estimate of the solution and upper and lower bounds on the variables are also passed to OPCON.

II.3 IMPLEMENTATION OF THE ALGORITHM

I.3.1 AN OVERVIEW

The following paragraphs describe in detail each of the steps performed by OPCON in roughly the order in which they occur. A brief overview of the algorithm is given first.

[0] Initialize. Initialize data structures and compute function values and finite difference estimates at the initial estimate of the solution.

[1] Start or Restart. Set the approximation to the Cholesky factor of the permuted Hessian of the Lagrangian to the identity. Set the multiplier estimates to zero. If it is a restart, then compute a finite difference interval for each variable using the procedure given by Gill, Murray, Saunders, and Wright [1981].

[2] Form the Dual QP. Form the dual to the quadratic program (QP) solved on each iteration of the SQP method. In forming the dual QP, consider only those inequality constraints that are active.

[3] Solve the Dual QP Using a CG Method. Solve for the multipliers of the original QP by solving the dual QP that has only nonnegativity constraints on some of the variables using a projected preconditioned conjugate gradient (PPCG) method. If unable to converge, then restart.

[4] Compute the Step Direction. Let g_ℓ be the gradient of the Lagrangian function at the current estimate of the solution for the primal variables and the multipliers from the QP obtained in step [3]. Then the step direction, s^c, is the solution of $L_c L_c' s = -g_\ell$, where L_c is the current estimate of the Cholesky factor of the Hessian of the Lagrangian.

[5] Compute the Step Length. Find an acceptable step αs^c. The criterion for an acceptable step is a suitable reduction in the merit function of Han (see equation I.6) based on a local linear model of the problem. Set the new estimate of the solution, x^n, to $x^c + \alpha s^c$. If unable to find an acceptable step, then restart.

[6] Check for Termination. Compute new estimates of the gradients of the objective and nonlinear constraint functions. Check for termination. Successful termination occurs if the norm of the Lagrangian gradient is small.

[7] Update the Cholesky Factorization. Update the approximation to the Cholesky factor of the Hessian of the Lagrangian using the BFGS formula as given by Goldfarb [1976] but ignore any change to structural zeros in L. If the update is unacceptable, then restart. Otherwise, begin a new iteration by going to step [2].

The following sections discuss the actual implementation of these steps in detail.

II.3.2 INITIALIZATION

The initialization section of the code reads in the data file defining the problem and establishes the sparse storage structures for the Hessian factorization and the gradients. The maximum number of active constraints is computed based on the number of variables in the problem and the amount of storage allocated to the array that will be used to store data defining the dual quadratic program. This array is dimensioned to be very large, but it is used for storing other data during different steps of the algorithm.

The objective and constraint functions are evaluated at the initial point. The degree of infeasibility is computed using the following formula:

$$(1) \quad FEAS = \sum_{j=1}^{m} [g_j(x^c) - RHS_j]_+ + \sum_{k=m+1}^{m+p} |h_k(x^c) - RHS_k|$$

$$+ \sum_{i=1}^{n} \max (0.0, x_i^l - x_i^c , x_i^c - x_i^u) ,$$

where the first m constraints are inequality constraints and the last m+1, ..., m+p constraints are equality constraints. The upper and lower bounds on the i-th variable are given by x_i^u and x_i^l, respectively. For j = i, ..., m+p, RHS_j specifies the right-hand-side value for constraint j.

The initial finite difference interval is set to an input value and is used for all variables. Forward differencing is used initially.

II.3.3 FIRST ITERATION AND ALL RESTARTS

First Iteration

On the first iteration and on all restarts, the Cholesky factor is set to the identity and all multiplier estimates are set to zero. The value of MAXLAM, which is used as an upper bound on the largest multiplier in absolute value, is also set to zero.

Restart

On each restart the actions described in the preceding section are repeated and in addition the following actions are taken. The finite difference interval for each variable is recomputed using the method of Gill, Murray, Saunders, and Wright [1981] that optimizes the interval for each variable in order to make the finite difference estimates of the gradients as accurate as possible. The method also computes central differencing intervals. The function used is the Lagrangian function. (During a restart, the multipliers are not reset to zero until the finite difference intervals are recomputed.) The method of Gill, Murray, Saunders, and Wright is available as a subroutine called FDCALC.

Central differencing is used if FDCALC detects an error condition or if the following inequality is satisfied for any variable:

$$\zeta_i < 10^m \Phi_{ii} h_i,$$

where m is 5 for this implementation. This test is described in Stewart [1967]. Φ_{ii} is an approximation to the i-th diagonal element of the Hessian of the function computed by FDCALC, h_i is the optimum forward differencing interval computed for the i-th variable, and ζ_i is an estimate of the i-th component of the gradient also computed by FDCALC. If a gradient estimate for a variable is small relative to the

Hessian estimate, then central differencing should be used because the forward difference estimate is unlikely to be very accurate.

II.3.4 FORM THE DUAL QP

Recall from Part I that the step direction computed at each iteration, s^c, is the solution to the following quadratic program:

$$\min_{s} \quad \nabla f(x^c)'s + (1/2)s'L_cL_c's$$

$$\text{subject to} \quad \nabla g_j(x^c)'s + g_j(x^c) \leq 0.0, \quad j=1, \ldots, m,$$

$$\nabla h_k(x^c)'s + h_k(x^c) = 0.0, \quad k=m+1, \ldots, m+p,$$

$$x_i^l \leq x_i^c + s_i \leq x_i^u, \quad i=1, \ldots, n,$$

where L_c is the current approximation to the Cholesky factor of the Hessian Lagrangian. By transforming to a dual problem a much simpler quadratic program can be solved. In order to make the dimension of the dual problem as small as possible, only active or nearly active constraints are considered. Equality constraints are always included. Any upper- or lower- bound constraint on a variable, x_i, will be included if the constraint is violated or if the current value of the variable is within BNDV*$(x_i^u - x_i^l)$ units of either the upper or lower bound. BNDV is an input parameter and is typically 5×10^{-4}. An

inequality constraint is considered active if the constraint is violated or if it is within ACTV units of being violated, where ACTV is another input parameter and is chosen to fit the scaling of the problem. It is important to properly scale the constraints since this criterion is applied to all the inequality constraints.

The dual quadratic program (see section I.3.1) is

$$(2) \quad \min_{u} \ (1/2)u'M'Mu + u'q$$

$$\text{subject to} \quad u_j \geq 0.0,$$

where the i-th column of M corresponds to an inequality constraint. Let the active constraints as defined in the previous section be numbered from 1 to p', including any active upper- or lower- bound constraints. Then the j-th column of M is the solution, y, to $L_c y = \nabla g_j(x^c)$, where g_j can be either an equality or inequality constraint. This is a sparse triangular system of equations that is easily solved using a sparse forward substitution method. Vector q is given by $q = M't - g(x^c) + RHS$ where t is the solution to $L_c t = \nabla f(x^c)$ and g is the vector of

active constraints. Again, a sparse triangular system is solved to obtain the vector.

As long as M has full column rank, which will be the case as long as the gradients of the active constraints are linearly independent and L_c is nonsingular, M'M, the matrix defining the dual quadratic program, will be positive definite and a solution to (2) will exist. If the gradients of the active constraints are not linearly independent, which will be the case if there are more active constraints than variables in the nonlinear program, then (2) cannot be solved with the CG method described in the next section. The algorithm recognizes when there are more constraints than variables and takes the following action. A value of ε is added to each of the diagonal elements of M'M, where ε is a small (10^{-5}) positive number input to OPCON. This is a perturbation to the original problem, which allows the algorithm to continue making progress when far from a solution (see section I.3.1). It is expected that the number of active constraints near the solution will be equal to or less than the number of variables.

II.3.5 SOLVE THE DUAL QP USING THE PPCG METHOD

The basic method for finding the solution to the dual QP using a projected preconditioned conjugate gradient (PPCG) method is given in section I.3.3. The computer implementation follows the steps given in the description, with a few exceptions.

The discussion here refers to the following quadratic program:

$$(3) \quad \min_{u} \ (1/2)u'Ku + k'u$$

$$\text{subject to} \quad u_i > 0.0 \text{ for } i \in I_{INEQ}$$

where I_{INEQ} is the set of indices corresponding to the inequality constraints considered active in (I.1) and $K = M'M$ or $K = M'M + \varepsilon I$ and $k = q$.

The exceptions include using an error tolerance to determine whether the system is ill-conditioned or nearly singular, passing through the algorithm the first time using a weak optimality criterion followed by a restart with the required criterion on the norm of the residual, and scaling the K matrix to speed convergence.

Checking for Ill-Conditioning in K

The value of $u_J'K_{JJ}u_J$ is computed during each conjugate gradient iteration where the subscript, J, refers to restricting attention to a subset of the variables in (3). If $u_J'K_{JJ}u_J/u_J'u_J$ is small, then K is nearly singular or ill-conditioned. If this condition is detected, a flag is set and an attempt is made to solve the perturbed problem to be described shortly.

Scaling the Dual QP Matrix

Before the PPCG algorithm is initiated, the matrix K is scaled as $K_{SCAL} = SKS$, where S is a diagonal matrix with $S_{ii} = (|K_{.i}|_2)^{-1/2}$. The problem actually solved is

$$\min_{v} \quad (1/2)\ v'K_{SCAL}v + v'Sd$$

$$\text{subject to} \quad v_i > 0.0 \text{ for } i \in I_{INEQ}.$$

The solution, u*, is given by $u^* = Sv^*$, where v^* is the solution to the scaled problem. Tests on the residual norm are always made relative to the unscaled problem to ensure that the error tolerance is satisfied for the desired problem. The initial estimate of the solution to the scaled problem is set to $v^o = S^{-1} u^o$, where u^o is the last estimate of the multiplier vector corresponding to the currently active constraints.

Perturbing the Dual QP

If the algorithm is unable to converge after an input number of iterations or if K appears to be singular or ill-conditioned, a small value, ε, is added to each of the diagonal elements of K_{SCAL} and the algorithm is restarted with the multipliers reinitialized to the zero vector. The small value added is an input parameter and is usually on the order of 10^{-5}. This procedure has an effect similar to that

discussed in the steps taken to form the dual QP when there are more active constraints than variables.

If the CG algorithm fails to converge, OPCON will restart.

II.3.6 COMPUTE THE STEP DIRECTION

After computing the new estimate of the multipliers the new step direction, s^c, is obtained as the solution to the equation:

$$L_c L_c' s = -\{\nabla f(x^c) + \sum_{j=1}^{m} u_j \nabla g_j(x^c) + \sum_{k=m+1}^{m+p} v_k \nabla h_k(x^c) + \sum_{i=1}^{n} \pm w_i e_i\} .$$

Any multipliers associated with inequality constraints considered to be inactive are set to zero. The multipliers for an active upper- or lower-bound constraint on the variable x_i, w_i, will be positive if the upper bound is active and negative if the lower bound is active. This sparse set of equations is solved using the sparse code of George and Liu [1981].

II.3.6 COMPUTE THE STEP LENGTH

OPCON will not accept a step unless the step results in a sufficient decrease in the Han merit function (see section I.4). Let FEASO be the sum of infeasibilities as given by (1) at the current estimate of the solution, x^c. The current value of the merit function is then

$$\text{PHIO} = f(x^c) + \text{MAXLAM} \cdot \text{FEASO} ,$$

where MAXLAM is an upper bound on the maximum of the absolute values of the multipliers. For $0 < \alpha \leq 1$, evaluate (1) at $x^c + \alpha s^c$ and define the value as FEASN. The step, αs^c, will be accepted if

$$\text{PHIN} = f(x^c + \alpha s^c) + \text{MAXLAM} \cdot \text{FEASN} < \text{PHIO} + \sigma\alpha \cdot \text{PHISLP}.$$

The test is made less strict by setting $0.01 < \sigma \leq 0.5$. The slope of the merit function is approximated by PHISLP, which is computed as follows:

$$\text{PHISLP} = f(x^c)'s^c + \text{MAXLAM} \left\{ \sum_{j=1}^{m} G_j(x^c) + \sum_{k=m+1}^{m+p} H_k(x^c) + \sum_{i=1}^{n} S_i \right\} ,$$

where for inequality constraints (see section I.4)

$$G_j(x^c) = \begin{cases} 0: \ g_j(x^c) < RHS_j \text{ or } [g_j(x^c) = RHS_j \text{ and } \nabla g_j(x^c)'s^c < 0.0] \\ \\ \nabla g_j(x^c)'s^c: \text{ otherwise} \end{cases}$$

for equality constraints

$$H_k(x^c) = \begin{cases} -\nabla h_k(x^c)'s^c: \ h_k(x^c) < RHS_k \\ \nabla h_k(x^c)'s^c: \ h_k(x^c) > RHS_k \\ \left|\nabla h_k(x^c)'s^c\right|: \ h_k(x^c) = RHS_k \end{cases}$$

and for upper- and lower-bound constraints

$$S_i = \begin{cases} s_i^c: \ x_i^c > x_i^u \text{ or } [x_i^c = x_i^u \text{ and } s_i^c > 0.0] \\ \\ -s_i^c: \ x_i^c < x_i^l \text{ or } [x_i^c = x_i^l \text{ and } s_i^c < 0.0]. \end{cases}$$

The first trial value of α is 1 . If this is not an acceptable step, then α is reduced by a constant factor less than one. This procedure is repeated until an acceptable step is found or until the number of trial values exceeds an input value, usually 8 to 16. If an acceptable step is not found in the allotted number of iterations, the algorithm will check the smallest value of the merit function obtained during any of the iterations. If this value is smaller than the value of the merit function at the current estimate, the algorithm will take

the step corresponding to the smallest value, then the algorithm will restart. No step is taken that does not improve the merit function.

II.3.7 CHECK FOR TERMINATION

After a successful step has been taken, an approximation to the gradient of the Lagrangian is computed using the current estimates of the multipliers and new finite difference estimates of the gradients of the objective and constraint functions. If

$$\left|\nabla_x \ell(x^n, u^n, v^n)\right| / \max\{ 1, \left|f(x^n)\right| \}$$

is less than the tolerance specified for stopping, the algorithm will terminate.

The algorithm will also stop if the step taken has norm less than a specified value, usually 10^{-9}. The algorithm will terminate, indicating a failure if, the iteration after a restart results in a failure since another restart would result in the same failure.

II.3.8 UPDATE THE APPROXIMATION TO THE HESSIAN OF THE LAGRANGIAN

The updating of the approximation to the Hessian of the Lagrangian function follows the procedure described in section I.5, with the exceptions noted below.

Recall that $y = \nabla_x \ell(x^n, u^n, v^n) - \nabla_x \ell(x^c, u^n, v^n)$, where x^n and x^c are the new and old estimates of the solution, respectively, and u^n and v^n are the vectors of multipliers. Let $s = \alpha s^c$ be the actual step taken. The BFGS update will not be positive definite unless $s'y > 0$. We follow the suggestion of Powell [1978] of setting y to

$$\theta y + (1 - \theta)Hs ,$$

where $\theta = 0.8(s'Hs/(s'Hs-s'y))$ if $s'y < (0.2)s'Hs$ in the formula for the BFGS update of the approximation matrix H. This will guarantee that the update is positive definite even if, originally, s'y was close to zero.

If either $s'y/|s||y|$ or $s'Hs/|s|^2$ is small ($< 10^{-5}$), the algorithm will restart since these are signs that the Hessian approximation is not good. If $|y|/|\nabla_x \ell(x^n, u^n, v^n)|$ is small, there will be no update but the algorithm will not restart.

II.4 PERFORMANCE OF OPCON

II.4.1 TEST PROBLEMS

It is difficult to find in the mathematical programming literature large-scale nonlinear programming problems suitable for use as test problems. Yet the performance of a large-scale algorithm on small, but well-known, test problems is an important part of evaluating the algorithm. This section describes the performance of the OPCON algorithm when solving eleven test problems, seven of which appear in the literature. The other four were either created by the author or obtained from unpublished sources. Of these latter four problems, one has 32 variables and the other three each have 60 variables. They are considered helpful in assessing the performance of a large-scale nonlinear programming algorithm.

Two problems were obtained by adding a set of ten nonlinear and five linear constraints to two well-known unconstrained optimization problems (Buckley and Lenir [1983]). These two problems are highly nonlinear and non-convex. The interiors of several degenerate ellipsoids are excluded from the feasible region by some of the

constraints. Since the unconstrained minima are contained in the excluded region, the problems are strongly nonconvex. Since these problems were created for this paper, we are unable to compare their solutions with any solutions obtained with other algorithms.

Also included in the set of test problems is a weapon-allocation problem having a single linear constraint and non-negativity constraints on the variables; an economic model of OPEC oil prices that has 10 nonlinear constraints and 40 linear constraints. Each of the eleven test problems is described in Appendix A.

II.4.2 TEST RESULTS

The problems were run on a VAX 11/780 minicomputer using double precision for all noninteger computations. The compiler option that stores double precision numbers in a format allowing a dynamic range of 10^{-307} to 10^{+307} was selected. Table 1 summarizes the test results. The tolerance for successful termination was set to 10^{-5} for all problems. The CPU time (in seconds) is the execution time of the program for each problem and does not include compilation or linking time. The total number of function evaluations and CG iterations are given. The number of each type of constraint -- nonlinear, linear, and upper- or lower-bound -- is given, followed by the number considered active at the time the algorithm terminated. The norm of the Lagrangian gradient and the sum of infeasibilities are also given. The number of

TABLE 1. OPCON PERFORMANCE SUMMARY (SPARSE HESSIANS)
Termination Criterion: $|\nabla_x \ell(x, u)| < 10^{-4} \max (1.0, |f(x)|)$

No.	Problem name	NVAR	Constraints (#/# active) Non linear	Linear	Bound	Iter.	Func eval.	CG iter.	Lgr. tol.	Feas.	CG fail	Step fail	Re-starts	CPU time (sec)
1	Betts	5	0/0	3/3	0/0	9	60	36	2.2E-04	4.8E-15	0	0	0	1
2	Proctor(1)	5	6/2	0/0	10/3	6	42	30	4.0E-04	1.6E-11	0	0	0	1
3	US Steel No. 1-2	6	4/1	0/0	12/5	2	21	11	5.8E-05	6.2E-16	0	0	0	1
4	US Steel No. 3-4	6	4/1	0/0	12/5	17	236	160	3.3E-08	1.9E-14	0	1	1	4
5	Hexagon	9	12/9	0/0	2/1	10	114	126	2.6E-06	1.2E-09	0	0	0	5
6	Wong No. 2	10	5/3	3/3	0/0	14	179	86	2.0E-03	5.8E-07	0	0	0	3
7	Dembo No. 1	12	3/3	0/0	0/0	45	570	126	1.7E-03	1.0E-05	0	0	0	16
8	Coville No. 2	15	5/5	0/0	15/6	33	1,330	1,215	3.0E-03	4.6E-13	1	1	2	52
9	Weapon Alloc.	32	0/0	1/1	32/25	106	3,993	1,948	1.4E-02	1.0E-12	1	1	2	286
10	World Bank	60	10/10	40/40	60/0	7	248	246	7.3E-02	2.2E-06	0	0	0	83
11	Mod. Powell Sing.	60	10/7	5/1	120/0	172	10,565	1,038	5.7E-05	1.1E-13	0	0	0	596
12	Mod. Extros	60	10/4	5/1	120/1	385	42,451	1,886	2.4E-04	5.4E-12	0	3	3	1,700

(1) Termination criterion set to 10^{-6} to force better solution.

times the CG algorithm failed to converge and the number of times an improved value of the merit function could not be found are also given for each problem. The total number of restarts is also shown.

All eleven problems terminated successfully for the termination tolerance given above. Even though the OPCON algorithm is not specifically designed to solve problems having linear constraints, it did perform well on the oil price model and tolerably well on the weapon-allocation model.

As noted earlier, one of the problems that must be handled by the algorithm is the possibility of M'M being singular or nearly so due to having more active constraints than variables or gradients of active constraints that are nearly linearly dependent. The algorithm handles this problem quite well. The eigenvalues of M'M were computed for each iteration for all of the test problems. Condition numbers ranged up to 10^9. The ill-conditioning seldom caused the CG algorithm to fail. The accuracy of the multiplier vector obtained under such circumstances is questionable; however, the algorithm almost never failed to take a step after solving these ill-conditioned problems.

It is interesting to note that the average number of CG iterations per main iteration for each problem is only slightly more than the number of active constraints (the order of the dual QP solved by each call to the CG algorithm) indicating the efficiency of solving the dual

and starting with the last estimate of the dual variables. The number of function evaluations seems high until one considers that finite differencing was used to compute all gradients. In some cases central differencing was used, which increases the number of function evaluations.

II.4.3 PARAMETER VALUES

In the process of obtaining these test results, it was found that the values of several of the input parameters are particularly critical to the successful termination of the algorithm. The choice of 0.1 for the step-length parameter, σ, was found to be quite good for most problems. Smaller values sometimes allowed the algorithm to drift, while larger values tended to cause more step-length iteration failures and hence considerable more computational effort since each failure causes a restart.

The convergence criterion for the norm of the residual in the PPCG algorithm was normally set to 10^{-12}. Larger values produced estimates of the multiplier vector that were sometimes not accurate enough to obtain a good step direction, whereas smaller values caused more CG iteration failures with no compensating improvement in performance. The relaxation parameter for the SSOR preconditioning step was set to 1.3. This value gave good performance while larger values typically took more iterations of the CG algorithm to converge.

Best choices for the parameters, BNDV and ACTV, used to determine whether an upper- or lower-bound constraint or a general inequality constraint should be considered active are very much problem dependent. A choice of 1.0 for ACTV and 0.0005 for BNDV was normally acceptable, but for some problems different values were used. For example, ACTV was set to 0.05 for the Proctor-Gamble problem and to 0.5 for the Hexagon problem. The BNDV parameter was set to 0.005 for the weapon-allocation problem, to allow it to pick up zero allocations more quickly. If BNDV or ACTV is set to too large a value, constraints not active at the solution may continue to be considered active when the algorithm gets close to a solution. In this case, it is possible that more constraints will be active than there are variables in the problem, and the need to solve the perturbed problem near the solution may hinder convergence.

II.4.4 FULL HESSIANS VERSUS SPARSE HESSIANS

One issue of major concern during the development of this algorithm was the degree to which ignoring fill-in in the BFGS update would degrade the performance of the algorithm. Table 2 shows the results of comparing full and sparse Hessian matrices on a subset of the test problems. Fill-in was not ignored for the full Hessian runs. The table shows the degree of sparseness for each problem. The termination criterion was set to 10^{-6}, to force a more stringent comparison.

TABLE 2. SPARSE HESSIAN VERSUS FULL HESSIAN
Termination Criterion: $|\nabla_x \ell(x, u)| < 10^{-6} \max\{1.0, |f(x)|\}$

No.	Problem Name	NVAR	Hessian density (%)	Iter.	Func. eval.	CG iter.	Lgr. tol.	Feas.	CG fail	Step fail	Re-starts	CPU time (sec)
4	US Steel No. 3-4	6	56	15	217	143	2.4E-09	3.6E-14	0	0	1	5
4	US Steel No. 3-4 F.H.(1)	6	100	17	440	171	4.2E-09	1.2E-14	0	1	3	7
5	Hexagon	9	70	11	124	135	2.6E-08	2.4E-13	0	0	0	5
5	Hexagon F.H.	9	100	11	128	99	2.5E-08	5.4E-10	0	0	0	5
6	Wong No. 2	10	12	19	234	121	1.4E-05	7.7E-10	0	0	0	5
6	Wong No. 2 F.H.	10	100	15	192	106	1.7E-05	1.3E-10	0	0	0	5
7	Dembo No. 1	12	93	300(2)	5,557	1,179	3.3E-04	0.0	0	1	1	134
7	Dembo No. 1 F.H.	12	100	300(2)	6,012	1,131	8.3E-05	0.0	0	1	1	138
8	Colville No. 2	15	16	46	1,593	735	1.5E-05	9.8E-14	1	0	1	56
8	Colville No. 2 F.H.	15	100	47	1,870	1,457	4.8E-06	2.6E-14	2	1	3	91

(1) Full Hessian.
(2) Stopped after 300 iterations without satisfying the termination criteria.

The value of this comparison is limited, considering the size of the test problems used. The results suggest, however, that the practical penalty for ignoring fill-in is minimal.

II.4.5 COMPARING OPCON TO MINOS 5.0

MINOS 5.0 is a well-known implementation of the projected Lagrangian algorithm developed by Murtagh and Saunders [1982]. The interested reader should consult the user's guide for MINOS 5.0 (Murtagh and Saunders [1983]) to obtain a full description of the features of the code. Performance of OPCON and MINOS 5.0 on several of the test problems is summarized in table 3. (Timing data for MINOS was obtained from the same VAX 11/780 system described earlier.)

MINOS is clearly superior to the current version of OPCON for problems having nearly linear constraints, such as the weapon-allocation problem and the World Bank problem. For problems that are highly nonlinear, especially in the constraints, OPCON is as good as MINOS and often much better. MINOS, for example, was unable to achieve any significant progress on the modified Powell singular function problem, whereas OPCON manages to find a feasible solution having a much improved objective function value. The Colville no. 2 problem has considerable nonlinearity in the objective function and constraints, but ten of its

TABLE 3. OPCON VS. MINOS COMPARISON

No.	Problem name	OPCON Function evaluations	OPCON CPU time (sec)	MINOS Function evaluations	MINOS CPU time (sec)
3	US Steel No. 1	21	1	59	3
4	US Steel No. 4	236	4	181(1)	5
5	Hexagon	114	5	287	7
6	Wong No. 2	179	3	1,836	23
7	Dembo No. 1	570	16	3,945	1,711
8	Colville No. 2	1,130	52	678	15
9	Weapon Allocation	3,993	286	1,398	24
10	World Bank	248	83	1,214	27
11	Mod. Powell Sing.	10,565	596	(2)	

(1) MINOS found a worse solution than the one found by OPCON.
(2) MINOS was unable to solve this problem.

15 variables appear as linear variables in all constraints and the objective function. As a result of its explicit handling of linear variables, MINOS did outperform OPCON on this problem.

II.5 CONCLUSIONS

Based on the results given in the preceding section, the OPCON algorithm shows promise of being a practical tool for solving large-scale nonlinear programs. Obviously, problems with 60 variables are not large; however, the scarcity of problems in the literature having even half as many variables would indicate that these results are significant. Also, some of these test problems -- i.e., the U.S. Steel, Colville, and Dembo problems -- are very difficult to solve in spite of their smallness. It is hoped that the algorithm will soon be applied to larger problems that would allow a more realistic evaluation of its ability to solve large, sparse nonlinear programs.

Because it uses an active set strategy and solves a dual problem, the algorithm should be able to deal with large numbers of nonlinear or linear inequality constraints since the size of the dual problem is determined by the number of active constraints. The current version of the OPCON algorithm stores the M matrix -- which has n rows and p' columns, where n is the number of variables in the problem and p' is the number of active contraints -- in a dense format. This array is the limiting factor on size. It may be possible to store this array in a

sparse format (see section II.3.4).

The coded version of the algorithm described here is not as efficient as it could be. The calculation of the eigenvalues of the M'M matrix is, for instance, unnecessary. Also, the function evaluation routine evaluates the objective and all nonlinear constraint functions on each call. As discussed in section II.2, a more efficient version of the code would split the objective function evaluation off from the nonlinear constraint function evaluations. This would allow a more efficient use of the technique of Coleman and More [1982] for reducing the number of function evaluations required to obtain finite difference estimates of sparse gradients.

Several questions remain unanswered. How well the algorithm will work for solving really large, practical problems is probably the most interesting of these. Other questions of interest include determining the effects of the errors in the estimates of the gradients and the deviation from the BFGS update when fill-in in the Cholesky factor is ignored on the performance of the algorithm on large-scale problems. It would be comforting to know what conditions are required to guarantee the convergence of the projected CG algorithm described earlier.

The step-length control procedure is simple, and it may be possible to improve the performance of the algorithm by improving this

procedure. For instance, an adaptive procedure that would allow step-length parameter, α, to be greater than 1 could produce better results on problems having singular or nearly singular Hessians.

The development and testing of an extension of the sequential quadratic programming algorithm for solving large, sparse nonlinear programs has been presented here . The test results indicate that the algorithm has the potential to be a practical tool for solving problems having highly nonlinear objective and constraint functions.

REFERENCES

Beale, E. M. L. (1967). "An introduction to Beale's method of quadratic programming," in Nonlinear Programming (J. Abadie, ed.) pp. 132-205, North-Holland, Amsterdam.

Bertsekas, D. P. (1980). "Projected Newton methods for optimization problems with simple constraints," LIDS-R-1025, Massachusetts Institute of Technology, Cambridge, MA.

Bertsekas, D. P. (1981). "Augmented Lagrangian and differentiable exact penalty methods," LIDS-P-113, Massachusetts Institute of Technology, Cambrigde, MA.

Boggs, P. T. and J. W. Tolle (1980). "Merit functions for nonlinear programming problems," Operations Research and Systems Analysis Report, University of North Carolina, Chapel Hill, NC.

Boggs, P. T. and J. W. Tolle (1981). "An implementation of a quasi-Newton method for constrained optimization," Operations Research and Systems Analysis Technical Report No. 81-3, University of North Carolina, Chapel Hill, NC. (To appear in the SIAM Journal for Numerical Analysis.)

Boggs, P. T., J. W. Tolle, and P. Wang (1982). "On the local convergence of quasi-Newton methods for constrained optimization," SIAM Journal on Control and Optimization, 20, No. 2.

Buckley, A. and A. Lenir (1983). "QN-like variable storage conjugate gradients," Mathematical Programming, 27, pp. 155-175.

Chamberlain, R. M., E. Lemarechal, H. C. Pederson, and M. J. D. Powell (1979). "The watchdog technique for forcing convergence in algorithms for constrained optimization," paper presented at the Tenth International Symposium on Mathematical Programming.

Choi, K. K., E. J. Hang, J. W. Hou, and V. N. Sohoni (1983). "Pshenichny's linearization method for mechanical system optimization," Journal on Mechanisms, Transmissions, and Automation in Design, 105, pp. 97-103.

Coleman, T. F. and A. R. Conn (1982). "Nonlinear programming via an exact penalty function: asymptotic analysis," Mathematical Programming, 24, pp. 123-136.

Coleman, T. F. and J. J. More (1982). "Software for estimating sparse Jacobian matrices," ANL-82-37, Argonne National Laboratory, Argonne, IL.

Dantzig, G. B. (1963). Linear Programming and Extensions, Princeton University Press, Princeton, NJ.

Dembo, R. S. (1976). "A set of geometric programming test problems and their solutions," Mathematical Programming, 10, pp. 192-213.

Dembo, R. S., S. C. Eisenstat, and T. Steihaug (1982). "Inexact Newton methods," SIAM Journal of Numerical Analysis, 19, pp. 400-408.

Dennis, J. E. and J. J. More (1977). "Quasi-Newton methods, motivation and theory," SIAM Review, 19, No. 1.

Dennis, J. E. and R. B. Schnabel (1983). Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, NJ.

Fletcher, R. (1980). Practical Methods of Optimization, Vol. 1: Unconstrained Optimization, John Wiley and Sons, Chichester, England.

Fletcher, R. (1981). Practical Methods of Optimization, Vol. 2: Constrained Optimization, John Wiley and Sons, Chichester, England.

George, A. and J. W. Liu (1981). Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall, Englewood Cliffs, NJ.

Gill, P. E., W. Murray, and M. H. Wright (1981). Practical Optimization, Academic Press, London.

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright (1981). "A procedure for computing forward-difference intervals for numerical optimization," Technical Report SOL 81-25, Department of Operations Research, Stanford University, Stanford, CA.

Goldfarb, D. (1976). "Factorized variable metric methods for unconstrained optimization," Mathematics of Computation, 30, No. 136, pp. 796-811.

Hageman, L.A. and D. M. Young (1981). Applied Iterative Methods, Academic Press, New York.

Han, S. P. (1976). "Superlinearly convergent variable metric algorithms for general nonlinear programming problems," Mathematical Programming, 11, pp. 263-282.

Han, S. P. (1977). "A globally convergent method for nonlinear programming," Journal of Optimization Theory and Applications, 22, No. 3, pp. 297-309.

Hock, W. and K. Schittkowski (1981). Test Examples for Nonlinear Programming Codes, Springer-Verlag, Berlin, Germany.

Lasdon, L. S. and A. D. Waren (1980). "Survey of nonlinear programming applications," Operations Research, 28, No. 5, pp. 1029-1073.

Murtagh, B. A. and M. A. Saunders (1982). "A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints," Mathematical Programming Study 16, North-Holland Pub. Co., pp 84-117.

Murtagh, B. A. and M. A. Saunders (1983). "MINOS 5.0 user's guide," Tech. Report SOL 83-20, Department of Operations Research, Stanford University, Stanford, CA.

Nocedal, J. and M. Overton (1983). "Projected Hessian updating algorithm for nonlinear constrained optimization," Technical Report 95, Courant Institute for Mathematical Sciences, New York University, New York.

O'Leary, D. P. (1981). "A generalized conjugate gradient algorithm for solving a class of quadratic programming problems," in Large Scale Matrix Problems (Bjorck et al., ed.), North-Holland Publishing Co., New York.

Polyak, B. T. (1969). "The conjugate gradient method in extremal problems," USSR Computational Mathematics and Mathematical Physics, 9.

Powell, M. J. D. (1978). "The convergence of variable metric methods for nonlinearly constrained optimization calculations," in Nonlinear Programming 3 (Mangasarian et al., ed.), Academic Press, New York.

Pshenichny, B. N. (1970). "Algorithms for the general problem of mathematical programming," Kibernetika, 5, pp. 120-125.

Sandgren, E. (1977). "The utility of nonlinear programming algorithms," Purdue University, Ph.D Thesis.

Shanno, D. F. (1980). "On variable-metric methods for sparse Hessians," Mathematics of Computation, 34, No. 150, pp. 499-514.

Thapa, M. N. (1983). "Optimization of unconstrained functions with sparse Hessian matrices-quasi-Newton methods," Mathematical Programming, 25, No. 2, pp. 158-182.

Toint, Ph. L. (1977). "On sparse and symmetric matrix updating subject to a linear equation," Mathematics of Computation, 31, No. 140, pp. 954-961.

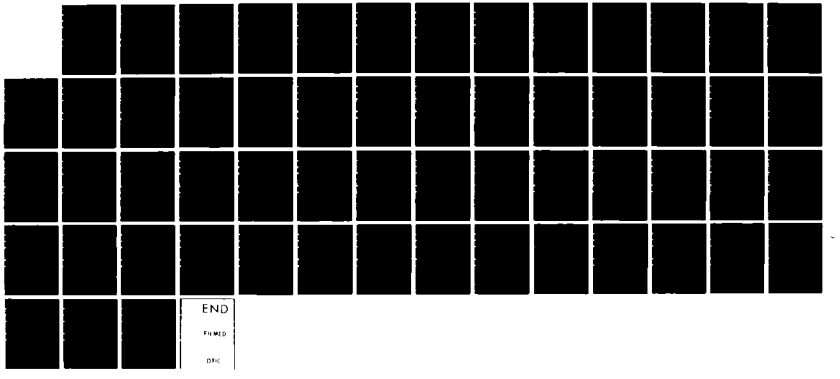
Waren, A. D. and L. S. Lasdon (1979). "The status of nonlinear programming software," Operations Research, 27, No. 3, pp. 431-456.

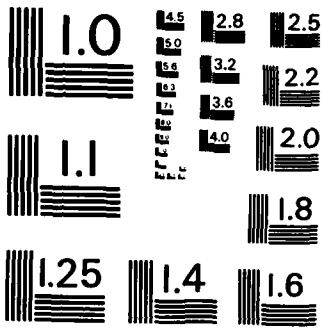
Wilson, R. B. (1963). "A simplicial method for convex programming," Ph.D. Dissertation, Harvard University, Cambridge, MA.

Wolfe, P. (1961). "A duality theorem for nonlinear programming," Quarterly of Applied Mathematics, 19, No. 3, pp. 239-244.

AD-A145 854 A SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM FOR SOLVING LARGE SPARSE NON.. (U) CENTER FOR NAVAL ANALYSES ALEXANDRIA VA AIR WARFARE RESEARCH D.. 2/2

UNCLASSIFIED R H NICKEL ET AL. MAY 84 CNA-PP-416 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

APPENDIX A

TEST PROBLEMS

The test problems used to evaluate the nonlinear optimization code developed for this paper are described in this appendix. Mathematical formulations are given for the problems having a small number of variables. A FORTRAN listing for the coded problem is included for each problem. Termination data reported for each problem includes the objective function value, $f(x^*)$, the norm of the gradient of the Lagrangian, $e(x^*)$, and the sum of the infeasibilities, $r(x^*)$. The starting points for each problem, the value of the objective function at the starting point, and the sum of the infeasibilities at the start are also reported. The best reported result by any other algorithm is also included.

[1] Betts Problem

Reference: Hock and Schittkowski [1981], problem no. 53

$$\min f(x) = (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2$$

subject to $x_1 + 3x_2 = 0,$

$x_3 + x_4 - 2x_5 = 0,$

$x_2 - x_5 = 0.$

Starting point: $x^0 = (7, 2, 6, 1, 2)'$,

$f(x^0) = 62.0,$

$r(x^0) = 16.0.$

Results:

	OPCON	Reported results
$f(x^*)$	4.093023256	4.093023269
$e(x^*)$	2.8E-10	2.17E-04
$r(x^*)$	0	4.83E-15

```
      SUBROUTINE FNCVAL(X,F)
C
C     SETT'S PROBLEM.
C
      DOUBLE PRECISION X(1),F(1),ONE,TWO
      DATA ONE,TWO/1.0D0,2.0D0/
C
      F(1)=(X(1)-X(2))**2+(X(2)+X(3)-TWO)**2+(X(4)-ONE)**2
     1       +(X(5)-ONE)**2
C
      RETURN
      END
```

[2] Proctor and Gamble Problem

Reference: Himmelblau [1972] problem no. 11

$$\min f(x) = 5.3578547\, x_3^2 + 0.8356891\, x_1 x_5 + 37.293239\, x_1$$

subject to

$$0 \leq 85.334407 + 0.0056858\, x_2 x_5 + 0.0006262\, x_1 x_4 - 0.0022053\, x_3 x_5 \leq 92$$

$$90 \leq 80.51249 + 0.0071317\, x_2 x_5 + 0.0029955\, x_1 x_2 + 0.0021813\, x_3^2 \leq 110$$

$$20 \leq 9.300961 + 0.0047026\, x_3 x_5 + 0.0012547\, x_1 x_3 + 0.0019085\, x_3 x_4 \leq 25$$

$$78 \leq x_1 \leq 102$$

$$33 \leq x_2 \leq 45$$

$$27 \leq x_3 \leq 45$$

$$27 \leq x_4 \leq 45$$

$$27 \leq x_5 \leq 45$$

Starting point: $x^o = (78.62, 33.44, 31.07, 44.18, 35.22)'$,

$f(x^o) = 10418.2$

$r(x^o) = 0.0.$

Results:

	OPCON	Reported results
f(x*)	10126.60285	10126.64100
e(x*)	4.05E-4	Not reported
r(x*)	1.60E-11	Not reported

```
      SUBROUTINE FNCVAL(X,F)
C
C     PROCTOR-GAMBLE CO. - HIMMELBLAU PROBLEM NO. 11.
C
      DOUBLE PRECISION X(1),F(1),A(3),B(4),C(4),D(4)
C
      DATA A/5.3578547,0.835689,37.293239/
      DATA B/85.334407,5.6858D-3,6.262D-4,-2.2053D-3/
      DATA C/80.51249,7.1317D-3,2.9955D-3,2.1813D-3/
      DATA D/9.30096,4.7026D-3,1.2547D-3,1.9085D-3/
C
      F(1)=A(1)*X(3)**2 +A(2)*X(1)*X(5) +A(3)*X(1)
C
      F(2)=B(1)+B(2)*X(2)*X(5)+B(3)*X(1)*X(4)+B(4)*X(3)*X(5)
      F(3)=-F(2)
      F(4)=C(1)+C(2)*X(2)*X(5)+C(3)*X(1)*X(2)+C(4)*X(3)**2
      F(5)=-F(4)
      F(6)=D(1)+D(2)*X(3)*X(5)+D(3)*X(1)*X(3)+D(4)*X(3)*X(4)
      F(7)=-F(6)
C
      RETURN
      END
```

[3] U.S. Steel Problems

References: Himmelblau [1972], problem no. 22; Hock and Schittkowski [1981], problem nos. 95-98.

$$\min f(x) = \sum_{i=1}^{6} c_i x_i$$

subject to

$$\sum_{i=1}^{6} a_{i1}x_i + b_1x_1x_3 + b_2x_3x_5 + b_3x_4x_5 + b_3x_4x_5 + b_4x_4x_6 + b_5x_5x_6 \leq B_1$$

$$\sum_{i=1}^{6} a_{i2}x_i + b_6x_1x_3 + b_7x_4x_5 + b_8x_4x_6 + b_9x_5x_6 \leq B_2$$

$$\sum_{i=1}^{6} a_{i3}x_i + b_{10}x_4x_5 \leq B_3$$

$$\sum_{i=1}^{6} a_{i4}x_i + b_{11}x_1x_6 \leq B_4$$

The [a..] and [b.] coefficients are given in the listing. Four problems are defined by the [B.]:

Problem	B_1	B_2	B_3	B_4
1	-4.97	+1.88	+29.08	+78.02
2	-4.97	+1.88	+69.08	+118.02
3	-32.97	-25.12	+29.08	+78.02
4	-32.97	-25.12	+124.08	+173.02

$0 \leq x_1 \leq 0.31$ $\qquad$ $0 \leq x_4 \leq 0.042$

$0 \leq x_2 \leq 0.046$ $\qquad$ $0 \leq x_5 \leq 0.028$

$0 \leq x_3 \leq 0.068$ $\qquad$ $0 \leq x_6 \leq 0.0134$

All four problems were solved by scaling the variables as follows: Scale x_1 by multiplying by 10; scale the other five variables by multiplying by 100.

Starting points:

For all problems $x^o = 0.0$.

For problem 2 $f(x^o) = 0.0$ and $r(x^o) = 4.97$.

For problem 4 $f(x^o) = 0.0$ and $r(x^o) = 58.09$.

Results:

Problems 1 and 2 have the same answer as do problems 3 and 4; therefore, only two sets of results are given.

	OPCON	Reported results
Problems 1-2		
f(x*)	0.01561952525	0.015619514
e(x*)	5.80E-05	0
r(x*)	6.17E-16	2.1E-09
Problems 3-4		
f(x*)	3.135809123	3.1358091
e(x*)	3.31E-09	0
r(x*)	1.89E-14	0

```
      SUBROUTINE FNCVAL(X,F)
C
C     US STEEL PROBLEM - HIMMELBLAU NO. 22.
C     SCALED VERSION.
C
      DOUBLE PRECISION X(1),F(1)
C
      INTEGER I,NVAR
      DOUBLE PRECISION A(6,4),B(11),C(6),DDOT,SCALE(6)
      DATA NVAR/6/
      DATA SCALE/1.0D1,5*1.0D2/
      DATA C/4.3D0,3.18D1,6.33D1,1.58D1,6.85D1,4.7D0/
      DATA A/-17.1D0,-38.2D0,-2.04D2,-2.123D2,-6.234D2,-1.4955D3,
     +       -17.9D0,-36.8D0,-1.139D2,-1.697D2,-3.378D2,-1.3852D3,
     +       0.D0,2.73D2,0.D0,7.0D1,8.19D2,0.D0,-1.599D2,3.11D2,
     +       0.D0,-5.87D2,-3.91D2,-2.198D3/
      DATA B/1.692D2,3.58D3,3.81D3,1.85D4,2.43D4,1.39D2,2.45D3,
     +       1.66D4,1.72D4,-2.6D4,1.4D4/
C
      DO 10 I=1,NVAR
         X(I)=X(I)/SCALE(I)
   10 CONTINUE
C
      F(1)=DDOT(NVAR,C,1,X,1)
      F(2)=DDOT(NVAR,A(1,1),1,X,1)+B(1)*X(1)*X(3)+B(2)*X(3)*X(5)+
     1          B(3)*X(4)*X(5)+B(4)*X(4)*X(6)+B(5)*X(5)*X(6)
      F(3)=DDOT(NVAR,A(1,2),1,X,1)+B(6)*X(1)*X(3)+B(7)*X(4)*X(5)+
     1          B(8)*X(4)*X(6)+B(9)*X(5)*X(6)
      F(4)=DDOT(NVAR,A(1,3),1,X,1)+B(10)*X(4)*X(5)
      F(5)=DDOT(NVAR,A(1,4),1,X,1)+B(11)*X(1)*X(6)
C
      DO 20 I=1,NVAR
         X(I)=SCALE(I)*X(I)
   20 CONTINUE
C
      RETURN
      END
```

[4] Hexagon Problem

References: Himmelblau [1972], problem no. 16; Hock and Schittkowski [1981], problem no. 108.

$$\min f(x) = -x_1x_4 + x_2x_3 - x_3x_4 + x_5x_4 - x_5x_8 + x_6x_7$$

subject to

$$x_3^2 + x_4^2 \leq 1$$

$$x_5^2 + x_6^2 \leq 1$$

$$x_1^2 + (x_2 - x_9)^2 \leq 1$$

$$(x_1 - x_5)^2 + (x_2 - x_6)^2 \leq 1$$

$$(x_1 - x_7)^2 + (x_2 - x_8)^2 \leq 1$$

$$(x_3 - x_5)^2 + (x_4 - x_6)^2 \leq 1$$

$$(x_3 - x_7)^2 + (x_4 - x_8)^2 \leq 1$$

$$x_7^2 + (x_8 - x_9)^2 \leq 1$$

$$x_2x_3 - x_1x_4 \leq 0$$

$$-x_3x_9 \leq 0$$

$$x_5x_9 \leq 0$$

$$x_6x_7 - x_5x_8 \leq 0$$

$$0 \leq x_9 \leq 1.$$

Starting point: $x^0_i = 1.0$, $i = 1, \ldots, 8$, $x_9^0 = .9$,

$f(x^0) = 0.0$,

$r(x^0) = 2.92$.

Results:

	OPCON	Reported result
f(x*)	-1.732050808	-1.732050808
e(x*)	2.63E-08	3.9E-10
r(x*)	2.37E-13	3.3E-12

```
      SUBROUTINE FNCVAL(X,F)
C
C     HIMMELBLAU PROBLEM NO. 16 - HEXAGON.
C
      DOUBLE PRECISION X(1),F(1)
C
      F(1)=X(1)*X(4)-X(2)*X(3)+X(3)*X(9)-X(5)*X(9)+X(5)*X(8)-X(6)*X(7)
      F(1)=-F(1)
C
      F(2)=X(3)**2+X(4)**2
      F(3)=X(5)**2+X(6)**2
      F(4)=X(1)**2+(X(2)-X(9))**2
      F(5)=(X(1)-X(5))**2+(X(2)-X(6))**2
      F(6)=(X(1)-X(7))**2+(X(2)-X(8))**2
      F(7)=(X(3)-X(5))**2+(X(4)-X(6))**2
      F(8)=(X(3)-X(7))**2+(X(4)-X(8))**2
      F(9)=X(7)**2+(X(8)-X(9))**2
      F(10)=X(2)*X(3)-X(1)*X(4)
      F(11)=-X(3)*X(9)
      F(12)=X(5)*X(9)
      F(13)=X(6)*X(7)-X(5)*X(8)
C
      RETURN
      END
```

[5] Wong Problem No. 2

Reference: Hock and Schittkowski [1981], problem no. 113.

$$\min f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45.0$$

subject to

$$3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 \leq 120$$

$$5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 \leq 40$$

$$0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 \leq 30$$

$$x_1^2 + 2(x_2 - 2)^2 + 2x_1x_2 + 14x_5 - 6x_6 \leq 0$$

$$-3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$$

$$4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 105$$

$$10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$$

$$-8x_1 + 2x_2 + 5x_9 - 2x_{10} \leq 12.$$

Starting point: $x^o_i = 0.0$, $i = 1, \ldots, 10$

$f(x^o) = 1352.0$

$r(x^o) = 810.0.$

Results:

	OPCON	Reported result
f(x*)	24.30620907	24.3062091
e(x*)	1.40E-05	1.2E-09
r(x*)	7.70E-10	4.6E-10

```
      SUBROUTINE FNCVAL(X,F)
C
      DOUBLE PRECISION X(1),F(1)
C
C     PROBLEM: WONG NO. 2; FROM H AND S #113 P. 122.
C
      DOUBLE PRECISION TEMP
C
      TEMP=X(1)**2+X(2)**2+X(1)*X(2)-14.0D0*X(1)-1.6D1*X(2)+
     $  (X(3)-1.0D1)**2+4.0D0*(X(4)-5.0D0)**2+(X(5)-3.0D0)**2+
     $  2.0D0*(X(6)-1.0D0)**2+5.0D0*X(7)**2+7.0D0*(X(8)-1.1D1)**2+
     $  2.0D0*(X(9)-1.0D1)**2+(X(10)-7.0D0)**2+4.5D1
      F(1)=TEMP
C
      TEMP=3.0D0*(X(1)-2.0D0)**2+4.0D0*(X(2)-3.0D0)**2+2.0D0*X(3)**2-
     $  7.0D0*X(4)-1.2D2
      F(2)=TEMP
C
      TEMP=5.0D0*X(1)**2+8.0D0*X(2)+(X(3)-6.0D0)**2-2.0D0*X(4)-4.0D1
      F(3)=TEMP
C
      TEMP=(5.0D-1)*(X(1)-8.0D0)**2+2.0D0*(X(2)-4.0D0)**2+
     $  3.0D0*X(5)**2-X(6)-3.0D1
      F(4)=TEMP
C
      TEMP=X(1)**2+2.0*(X(2)-2.0D0)**2-2.0D0*X(1)*X(2)+1.4D1*X(5)-
     $  6.0D0*X(6)
      F(5)=TEMP
C
      TEMP=-3.0D0*X(1)+6.0D0*X(2)+1.2D1*(X(9)-8.0D0)**2-7.0D0*X(10)
      F(6)=TEMP
C
      RETURN
      END
```

[6] Dembo's Problem no. 1B

References: Dembo [1976]; Sandgren [1977], problem no. 17

$$\min f(x) = 10^5 \prod_{i=1}^{11} x_i^{-a_i}$$

subject to

$$c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4x_5 \leq 1.0$$

$$c_5x_1 + c_6x_2 + c_7x_3 + c_8x_4x_{12} + c_9 \frac{x_5}{x_{12}} + c_{10} \frac{x_6}{x_{12}}$$
$$+ c_{11}x_7x_{12} + c_{12}x_4x_5 + c_{13} \frac{x_2x_5}{x_{12}} + c_{14}x_2x_4x_5$$
$$+ c_{15} \frac{x_2x_5}{x_4x_{12}} + c_{16} \frac{x_{10}}{x_{12}} \leq 1.0,$$

$$c_{17}x_1 + c_{18}x_2 + c_{19}x_3 + c_{20}x_4 + c_{21}x_5 + c_{22}x_6 + c_{23}x_8$$
$$+ c_{24}x_4x_5 + c_{25}x_2x_5 + c_{26}x_2x_4x_5 + c_{27} \frac{x_2x_5}{x_4}$$
$$+ c_{28}x_9 + c_{29}x_1x_9 + c_{30}x_{11} \leq 1.0,$$

$0.1 \leq x_i \leq 100.0 \quad i = 1, \ldots, 12.$

Starting point: $x^o_i = 4.0$, $i = 1, \ldots, 12$,

$f(x^o) = 0.22768$,

$r(x^o) = 15.114$.

This problem was solved by scaling the first 11 variables by multiplying each by 10.

Results:

	OPCON	Reported result
$f(x^*)$	3.169024101	3.1682133
$e(x^*)$	3.33E-04	--
$r(x^*)$	0.0	--

This is a difficult problem that OPCON has not handled very well. It is a geometric programming problem that is poorly scaled. OPCON required 300 iterations to achieve the results shown, and the optimal values for the variables are not as close as one would like to those reported by Dembo [1976].

```
      SUBROUTINE FNCVAL(X,F)
C
C     DEMBO PROBLEM NO. 1 FROM SANDGRENS THESIS.
C
      DOUBLE PRECISION X(1),F(1)
C
      DOUBLE PRECISION TEMP,TEMPX,EPS,ONE,TEN
      DOUBLE PRECISION B,A(11),C(30)
      DATA EPS,ONE,TEN/1.0D-15,1.0D0,1.0D1/
      DATA B,A/1.0D5,1.33172D-3,2.270927D-3,2.48546D-3,4.67D0,
     +   4.671973D0,8.14D-3,8.092D-3,5.D-3,9.09D-4,8.8D-4,1.19D-3/
      DATA C/5.367373D-2,2.1863746D-2,9.7733533D-2,6.6940803D-3,
     +   1.D-6,1.D-5,1.D-6,1.D-10,1.D-8,1.D-2,1.D-4,1.0898645D-1,
     +   1.6108052D-4,1.D-23,1.9304541D-6,1.D-3,1.D-6,1.D-5,1.D-6,
     +   1.D-9,1.D-9,1.D-3,1.D-3,1.0898645D-1,1.6108052D-5,1.D-23,
     +   1.930451D-8,1.D-5,1.1184059D-4,1.D-4/
C
      DO 5 I=1,11
         X(I)=X(I)/TEN
    5 CONTINUE
C
      TEMP=B
      DO 10 I=1,11
         IF (X(I).GT.EPS) THEN
            TEMPX=X(I)
         ELSE
            TEMPX=EPS
         ENDIF
         TEMP=TEMP*TEMPX**(-A(I))
   10 CONTINUE
      F(1)=TEMP
C
      F(2)=TEN*(C(1)*X(1)+C(2)*X(2)+C(3)*X(3)+C(4)*X(4)*X(5)-ONE)
C
      F(3)=TEN*(C(5)*X(1)+C(6)*X(2)+C(7)*X(3)+C(8)*X(4)*X(12)+
     +     C(9)*X(5)/X(12)+C(10)*X(6)/X(12)+C(11)*X(7)*X(12)+
     +     C(12)*X(4)*X(5)+C(13)*X(2)*X(5)/X(12)+
     +     C(14)*X(2)*X(4)*X(5)+C(15)*X(2)*X(5)/(X(4)*X(12))+
     +     C(16)*X(10)/X(12)-ONE)
C
      F(4)=TEN*(C(17)*X(1)+C(18)*X(2)+C(19)*X(3)+C(20)*X(4)+C(21)*X(5)+
     +     C(22)*X(6)+C(23)*X(8)+C(24)*X(4)*X(5)+C(25)*X(2)*X(5)+
     +     C(26)*X(2)*X(4)*X(5)+C(27)*X(2)*X(5)/X(4)+C(28)*X(9)+
     +     C(29)*X(1)*X(9)+C(30)*X(11)-ONE)
C
      DO 20 I=1,11
         X(I)=TEN*X(I)
   20 CONTINUE
C
      RETURN
      END
```

[7] Colville's Problem No. 2

References: Sandgren [1977], problem no. 14; Hock and Schittkowski [1981], problem no. 117; Himmelblau [1972], problem no. 18.

$$\min f(x) = \sum_{i=1}^{10} b_i x_i + \sum_{j=11}^{15} \sum_{i=11}^{15} c_{i-10,j-10}\, x_i x_j + 2 \sum_{j=11}^{15} d_{j-10}\, x_j^3$$

subject to

$$-2 \sum_{i=11}^{15} c_{i-10,j}\, x_i + \sum_{i=1}^{10} a_{ij} x_i - 3 d_j x_{10+j}^2 \leq 0.0_j, \quad j = 1, \ldots, 5,$$

$$x_j \geq 0, \quad j = 1, \ldots, 15.$$

See the listing for coefficients.

Starting point: $x^0_j = 0.001$, $j = 1, \ldots, 15$, $j \neq 7$;

$x^0_7 = 60.0$,

$f(x^0) = 2400.1053$,

$r(x^0) = 0.0$.

Results:

	OPCON	Reported result
f(x*)	32.34867897	32.348679
e(x*)	2.21E-05	3.5E-05
r(x*)	6.20E-14	0.0

```
      SUBROUTINE FNCVAL(X,F)
C
C     COLVILLES PROBLEM NO. 2
C
      DOUBLE PRECISION X(1),F(1)
C
      INTEGER I,J
      DOUBLE PRECISION ZERO
      DOUBLE PRECISION B(10),C(5,5),D(5),A(10,5),B1,C1,C2,D1
      DOUBLE PRECISION RHS(5)
      DATA B/4.D1,2.D0,2.5D-1,4.D0,4.D0,1.D0,4.D1,6.D1,-5.D0,-1.D0/
      DATA D/4.D0,8.D0,1.D1,6.D0,2.D0/
      DATA C/3.D1,-2.D1,-1.D1,3.2D1,-1.D1,-2.D1,3.9D1,-6.D0,-3.1D1,
     +       3.2D1,-1.D1,-6.D0,1.D1,-6.D0,-1.D1,3.2D1,-3.1D1,-6.D0,
     +       3.9D1,-2.D1,-1.D1,3.2D1,-1.D1,-2.D1,3.D1/
      DATA A/-1.6D1,0.D0,-3.5D0,0.D0,0.D0,2.D0,-1.D0,-1.D0,1.D0,1.D0,
     +       2.D0,-2.D0,0.D0,-2.D0,-9.D0,0.D0,-1.D0,-2.D0,2.D0,1.D0,
     +       0.D0,0.D0,2.D0,0.D0,-2.D0,-4.D0,-1.D0,-3.D0,3.D0,1.D0,
     +       1.D0,4.D0,0.D0,-4.D0,1.D0,0.D0,-1.D0,-2.D0,4.D0,1.D0,
     +       0.D0,2.D0,0.D0,-1.D0,-2.8D0,0.D0,-1.D0,-1.D0,5.D0,1.D0/
      DATA RHS/15.0,27.0,36.0,18.0,12.0/
      DATA ZERO/0.D0/
C
      B1=ZERO
      DO 10 I=1,10
         B1=B1+B(I)*X(I)
   10 CONTINUE
C
      C1=ZERO
      DO 30 J=11,15
         DO 20 I=11,15
            C1=C1+C(I-10,J-10)*X(I)*X(J)
   20    CONTINUE
   30 CONTINUE
C
      D1=ZERO
      DO 40 J=11,15
         D1=D1+D(J-10)*X(J)**3
   40 CONTINUE
C
      F(1)=B1+C1+2.D0*D1
C
      DO 100 J=1,5
         C1=ZERO
         DO 50 I=11,15
            C1=C1+C(I-10,J)*X(I)
   50    CONTINUE
C
         C2=ZERO
         DO 60 I=1,10
            C2=C2+A(I,J)*X(I)
   60    CONTINUE
C
         F(J+1)=RHS(J)-2.D0*C1 + C2 - 3.D0*D(J)*X(J+10)**2
```

```
  100 CONTINUE
C
      RETURN
      END
```

[8] Weapon-Allocation Problem

References: None.

$$\min - \sum_{j=1}^{6} n_j \prod_{i=1}^{7} (1 - p_{ij})^{x_{ij}/nj}$$

$$\text{subject to} \quad \sum_{i=1}^{7} c_i \sum_{j=1}^{6} x_{ij} \leq 4900$$

$$x_{ij} \geq 0, \; i = 1, \ldots, 7, \; j = 1, \ldots, 6.$$

Data:

p_{ij}

Weapons

	i/	1	2	3	4	5	6	7	n_j
	j								
Targets	1	.50	.58	.42	.42	0	0	0	5
	2	.30	.31	.37	.36	.19	0	0	40
	3	.10	.12	.20	.30	0	0	0	55
	4	.05	.05	.07	.07	0	.40	.45	18
	5	.68	.68	.68	.61	.77	.59	.90	18
	6	.43	.43	.35	.29	.41	.75	0	70
	c_i	12	12	12	15.6	21.6	3.5	21.3	

This is a weapon-allocation problem where x_{ij} denotes the number of weapons of type i to be allocated to the class of targets of type j. The objective function is the negative of a utility for a given allocation. The constraint is a volume constraint on storage. It would also be a cost constraint. The variables corresponding to p_{ij} being zero are not considered in the optimization, so there are only 32 variables in the problem.

Start: $x_{ij} = 2.0$ for all i and j,

$f(x^0) = -29.535$,

$r(x^0) = 0.0$.

Result:

	OPCON
$f(x^*)$	-167.7054586
$e(x^*)$	1.37E-02
$r(x^*)$	1.02E-12

This problem is difficult because it takes a long time to determine which variables are nonzero (eight are nonzero at the solution). Also, convergence is slow.

```
      SUBROUTINE FNCVAL(X,F)
C
C     THIS IS A WEAPONS ALLOCATION PROBLEM.
C
      DOUBLE PRECISION X(1),F(1)
C
      INTEGER TARGET(32),TARDEX,I,J
      DOUBLE PRECISION PK(32),TEMP,PKCUM(6),NTAR(6)
C
      DATA NTAR/5.0,40.0,55.0,18.0,18.0,70.0/
      DATA PK/.50,.30,.10,.05,.68,.43,.58,.31,.12,.05,.68,.43,.42,.37,
     +        .20,.07,.68,.35,.42,.36,.30,.07,.61,.29,.19,.77,.41,.40,
     +        .59,.75,.45,.90/
      DATA TARGET/1,2,3,4,5,6,1,2,3,4,5,6,1,2,3,4,5,6,1,2,3,4,5,6,2,5,6,
     +       4,5,6,4,5/
C
      DATA ZERO,ONE/0.0D0,1.0D0/
C
      DO 10 J=1,6
         PKCUM(J)=ONE
   10 CONTINUE
C
      DO 20 I=1,32
         TARDEX=TARGET(I)
         PKCUM(TARDEX)=PKCUM(TARDEX)*(ONE-PK(I))**(X(I)/NTAR(TARDEX))
   20 CONTINUE
C
      TEMP=ZERO
      DO 30 J=1,6
         TEMP=TEMP+NTAR(J)*(PKCUM(J)-ONE)
   30 CONTINUE
C
      F(1)=TEMP
C
      RETURN
      END
```

[9] World Bank Oil Price Model

References: None (this problem was obtained from A. Drud of the World Bank).

Let

p_i = oil price in year i/10.0,

t_{di} = total demand for oil in year i/10.0,

s_i = supply of oil by non-OPEC countries in year i,

cs_i = cumulative supply by non-OPEC countries in year i/10.0,

d_i = demand for OPEC oil in year i/10.0, and

r_i = OPEC reserves of oil in year i/100.0.

OPEC oil revenue in year i is given by

$$10.0\ d_i(10.0\ p_i - 2.5/r_i).$$

OPEC wants to maximize discounted oil revenue over 10 years.

$$\min f(x) = -\sum_{i=1}^{10} 10.0\ d_i(10.0\ p_i - 2.5/r_i)\left(\frac{1}{1.05}\right)^{i-1}$$

subject to

$$10.0\ td_i - 8.7td_{i-1} + 1.3p_i - 1.0 - 2.3\left(\frac{1}{1.015}\right)^{i-1} = 0,$$
$$i = 1, \ldots, 10$$

$$s_i - .75\ s_{i-1} - (1.1 + p_i)^{-cs_i/7.0} = 0,\ i = 1, \ldots, 10$$

$$10.0\ cs_i - 10.0\ cs_{i-1} - s_i = 0,\ i = 1, \ldots, 10$$

$$10.0\ d_i - td_i + s_i = 0,\ i = 1, \ldots, 10$$

$$10.0\ r_i - 10.0\ r_{i-1} + d_i = 0,\ i = 1, \ldots, 10$$

$$p_i,\ td_i,\ s_i,\ cs_i,\ d_i,\ r_i \geq 0,\ i = 1, \ldots, 10.$$

This problem has 60 variables and 50 equality constraints.

Initial values for year 0: $td_o = 1.8$,
$s_o = 6.5$,
$cs_o = 0.0$,
$r_o = 5.0$.

Start:
$$p_i = 1.4,\ i = 1, \ldots, 10$$
$$td_i = 1.8,\ i = 1, \ldots, 10$$
$$s_i = 7.0,\ i = 1, \ldots, 10$$
$$d_i = td_i - s_i,\ i = 1, \ldots, 10$$
$$cs_i = \sum_{j=1}^{i} s_j,\ i = 1, \ldots, 10$$

$r_i = r_{i-1} - d_i$, $i = 1, \ldots, 10$

$f(x^o) = -1198.1202$

$r(x^o) = 11.565.$

Results:

	OPCON	World Bank Optimization (CONOPT)
$f(x^*)$	-818.4235909	-818.42359
$e(x^*)$	5.34E-02	--
$r(x^*)$	3.23E-07	--

```
      SUBROUTINE FNCVAL(XDUM,FDUM)
C
C     WORLD BANK ECONOMIC MODEL OF OIL PRICES.
C     TEN YEAR HORIZON.
C     SCALED VERSION.
C
      DOUBLE PRECISION X(60),F(11),XDUM(1),FDUM(1)
      DOUBLE PRECISION P(10),S(10),CS(10),D(10),R(10),SEQ(10)
      EQUIVALENCE (X(1),P),(X(21),S),(X(31),CS),(X(41),D),
     +            (X(51),R),(F(2),SEQ)
      DOUBLE PRECISION S0,C(7),ZERO,ONE,TEN
      DATA ZERO,ONE,TEN/0.0D0,1.0D0,1.0D1/
      DATA S0/6.5D0/
      DATA C/2.5D0,1.05D0,.75D0,1.1D0,1.D0,1.02D0,7.0D-1/
C
C     TRANSFER INPUT VALUES TO SUBROUTINE VARIABLES.
C
      DO 100 I=1,60
         X(I)=XDUM(I)
  100 CONTINUE
C
      F(1)=ZERO
      DO 200 I=1,10
         F(1)=F(1)-TEN*D(I)*(TEN*P(I)-C(1)/R(I))*C(2)**(1-I)
         IM1=I-1
         IF (I.EQ.1) THEN
            SEQ(I)=S(I)-C(3)*S0-(C(4)+C(5)*P(I))*C(6)**(-CS(I)/C(7))
         ELSE
            SEQ(I)=S(I)-C(3)*S(IM1)-(C(4)+C(5)*P(I))*
     $                        C(6)**(-CS(I)/C(7))
         ENDIF
  200 CONTINUE
C
C     TRANSFER LOCAL FUNCTION VALUES TO DUMMY OUTPUT VARIABLE.
C
      DO 300 I=1,11
         FDUM(I)=F(I)
  300 CONTINUE
C
      RETURN
      END
```

[10] Modified Powell Singular Function Problem

References: Buckley and Lenir [1983]

This is a modified version of Powell's singular function with constraints added to create a large, constrained problem.

$$\min f(x) = \frac{1}{1000} \sum_{j=1}^{15} [c_1(x_{4j-3} + 10x_{4j-2})^2 + 5(c_2 x_{4j-1} - x_{4j})^2$$

$$+ c_3(x_{4j-2} - 2x_{4j-1})^4 + 10(x_{4j-3} - c_4 x_{4j})^4]$$

subject to (see the listing that follows for the code for the objective function and 10 nonlinear constraints; the linear constraints appear below)

$$-10x_{15} - 2x_{25} - 5x_{32} \leq -10$$

$$-2x_{11} - 5x_{41} - 10x_{57} \leq -10$$

$$-5x_{13} - 10x_{24} - 2x_{60} \leq -15$$

$$-5x_{17} - 2x_{33} - 2x_{41} \leq -20$$

$$-5x_{28} - 6x_{31} - 20x_{55} - 3x_{56} \leq -10$$

$$-10.0 \leq x_i \leq 10.0, \; i = 1, \ldots, 60.$$

Start: $x^o_i = 1.0,\ i = 1, \ldots, 60.$

$f(x^o) = 1.868,$

$r(x^o) = 13.85.$

Result:

	OPCON
$f(x^*)$	0.9490162422
$e(x^*)$	5.69E-05
$r(x^*)$	1.07E-13

```
      SUBROUTINE FNCVAL(X,F)
C
C     A CONSTRAINED VERSION OF THE POWELL SINGULAR FUNCTION
C     MINIMIZATION PROBLEM.
C
      DOUBLE PRECISION X(1),F(1)
C
      INTEGER J
      DOUBLE PRECISION TEMP,TWO,FIVE,TEN,ZERO
      DOUBLE PRECISION C(10)
C
      DATA ZERO,TWO,FIVE,TEN/0.0,2.0,5.0,10.0/
      DATA C/1.02,.905,1.08,.963,2.05,.895,1.325,.974,1.114,1.009/
C
      TEMP=ZERO
      DO 10 J=1,15
         TEMP=TEMP+C(1)*(X(4*J-3)+TEN*X(4*J-2))**2 +
     $              FIVE*(C(2)*X(4*J-1)-X(4*J))**2 +
     $              C(3)*(X(4*J-2)-TWO*X(4*J-1))**4 +
     $              TEN*(X(4*J-3)-C(4)*X(4*J))**4
   10 CONTINUE
C
      F(1)=TEMP/1.0D3
C
      TEMP=ZERO
      DO 20 J=1,20
         TEMP=TEMP-C(5)*(X(3*J-2)+TWO*X(3*J-1))**2 -
     $              C(6)*(X(3*J-1)+FIVE*X(3*J))**2 -
     $              TEN*(C(7)*X(3*J-2)-X(3*J))**2
   20 CONTINUE
C
      F(2)=(TEMP+5.0D2)/1.0D3
C
      TEMP=ZERO
      DO 30 J=1,6
         TEMP=TEMP-C(8)*(X(10*J)+X(10*J-9))**2 -
     $              C(9)*(TWO*X(10*J-5)-X(10*J-9))**2 -
     $              FIVE*(C(10)*X(10*J)+TWO*X(10*J-5))**2
   30 CONTINUE
C
      F(3)=(TEMP+4.0D2)/1.0D3
C
      TEMP=ZERO
      DO 40 J=1,3
         TEMP=TEMP-C(1)*(X(20*J)-X(20*J-19))**2 -
     $              C(2)*(X(20*J)+X(20*J-7))**2 -
     $              C(3)*(X(20*J-7)-TWO*X(20*J-19))**2
   40 CONTINUE
C
      F(4)=(TEMP+4.5D2)/1.0D3
C
      TEMP=ZERO
      DO 50 J=1,6
         TEMP=TEMP-C(4)*(X(10*J-1)+TWO*X(10*J-8))**2 -
```

```
     $              (C(5)*X(10*J-1)-FIVE*X(10*J-6))**2 -
     $              C(6)*(X(10*J-8)+TEN*X(10*J-6))**2
   50 CONTINUE
C
      F(5)=(TEMP+7.0D2)/1.0D3
C
      TEMP=ZERO
      DO 60 J=1,10
         TEMP=TEMP-C(7)*(X(6*J)-TWO*X(6*J-5))**2 -
     $              FIVE*(X(6*J)+C(8)*X(6*J-3))**2 -
     $              (C(9)*X(6*J-5)-TWO*X(6*J-3))**2
   60 CONTINUE
C
      F(6)=(TEMP+1.0D3)/1.0D3
C
      TEMP=ZERO
      DO 70 J=1,3
         TEMP=TEMP-C(10)*(X(20*J-4)+TWO*X(20*J-16))**2 -
     $              TWO*(C(1)*X(20*J-4)-X(20*J-10))**2 -
     $              (X(20*J-10)-C(2)*X(20*J-16))**2
   70 CONTINUE
C
      F(7)=(TEMP+1.0D3)/1.0D3
C
      TEMP=ZERO
      DO 80 J=1,12
         TEMP=TEMP-(C(3)*X(5*J)-TWO*X(5*J-2))**2 -
     $              TEN*(X(5*J)+C(4)*X(5*J-4))**2 -
     $              (X(5*J-4)+X(5*J-2))**2
   80 CONTINUE
C
      F(8)=(TEMP+1.0D3)/1.0D3
C
      F(9)=(-(C(5)*X(10)-TWO*X(11))**2 - X(10)*X(11) + 2.0D1)/1.0D3
C
      F(10)=(-(C(6)*X(50)-TEN*X(40))**2 - X(50)*X(40) + 5.0D1)/1.0D3
C
      F(11)=(-(C(7)*X(21)+X(22)+C(8)*X(23))**2 + 7.0D1)/1.0D3
C
      RETURN
      END
```

[11] Modified EXTROS Function Problem

Reference: Buckley and Lenir [1983]

This is a modified version of the EXTROS function described in the reference. The constraint functions are similar to those for the preceding problem.

$$\min f(x) = \frac{1}{100} \sum_{i=1}^{30} [100 (x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2]$$

subject to (see the listing that follows and the linear constraints for the preceding problem).

Start: $x^o_i = 1.0$, $i = 1, \ldots, 60$

$f(x^o) = 0.0$

$r(x^o) = 1238.9.$

Result:

	OPCON
$f(x^*)$	0.1289735371
$e(x^*)$	9.78E-05
$r(x^*)$	0.0

```
      SUBROUTINE FNCVAL(X,F)
C
C     A CONSTRAINED VERSION OF THE EXTROS FUNCTION
C     MINIMIZATION PROBLEM.
C
      DOUBLE PRECISION X(1),F(1)
C
      INTEGER J
      DOUBLE PRECISION TEMP,ONE,TWO,FIVE,TEN,HUNDRD,ZERO
      DOUBLE PRECISION C(24)
C
      DATA ZERO,ONE,TWO,FIVE,TEN,HUNDRD/0.D0,1.D0,2.D0,5.D0,1.D1,1.0D2/
      DATA C/1.02,.905,1.08,.963,2.05,.895,1.325,.974,1.114,1.009,
     $   2.57,8.04,-6.237,4.109,5.9599,6.2,-2.1,8.39,5.39,8.621,
     $   5.0432,7.211,4.903,1.327/
C
      TEMP=ZERO
      DO 10 J=1,30
         TEMP=TEMP+HUNDRD*(X(2*J)-X(2*J-1)**2)**2+(ONE-X(2*J-1))**2
   10 CONTINUE
C
      F(1)=TEMP/HUNDRD
C
      TEMP=ZERO
      DO 20 J=1,20
         TEMP=TEMP-C(1)*(X(3*J-2)+TWO*X(3*J-1))**2 -
     $             C(2)*(X(3*J-1)+FIVE*X(3*J))**2 -
     $             TEN*(C(3)*X(3*J-2)-X(3*J))**2
   20 CONTINUE
C
      F(2)=TEMP+5.0D2
C
      TEMP=ZERO
      DO 30 J=1,6
         TEMP=TEMP-C(4)*(X(10*J)+X(10*J-9))**2 -
     $             C(5)*(TWO*X(10*J-5)-X(10*J-9))**2 -
     $             FIVE*(C(6)*X(10*J)+TWO*X(10*J-5))**2
   30 CONTINUE
C
      F(3)=TEMP+4.0D2
C
      TEMP=ZERO
      DO 40 J=1,3
         TEMP=TEMP-C(7)*(X(20*J)-X(20*J-19))**2 -
     $             C(8)*(X(20*J)+X(20*J-7))**2 -
     $             C(9)*(X(20*J-7)-TWO*X(20*J-19))**2
   40 CONTINUE
C
      F(4)=TEMP+4.5D2
C
      TEMP=ZERO
      DO 50 J=1,6
         TEMP=TEMP-C(10)*(X(10*J-1)+TWO*X(10*J-8))**2 -
     $             (C(11)*X(10*J-1)-FIVE*X(10*J-6))**2 -
```

```
     $              C(12)*(X(10*J-8)+TEN*X(10*J-6))**2
  50 CONTINUE
C
      F(5)=TEMP+7.0D2
C
      TEMP=ZERO
      DO 60 J=1,10
         TEMP=TEMP-C(13)*(X(6*J)-TWO*X(6*J-5))**2 -
     $              FIVE*(X(6*J)+C(14)*X(6*J-3))**2 -
     $              (C(15)*X(6*J-5)-TWO*X(6*J-3))**2
  60 CONTINUE
C
      F(6)=TEMP+1.0D3
C
      TEMP=ZERO
      DO 70 J=1,3
         TEMP=TEMP-C(16)*(X(20*J-4)+TWO*X(20*J-16))**2 -
     $              TWO*(C(17)*X(20*J-4)-X(20*J-10))**2 -
     $              (X(20*J-10)-C(18)*X(20*J-16))**2
  70 CONTINUE
C
      F(7)=TEMP+1.0D3
C
      TEMP=ZERO
      DO 80 J=1,12
         TEMP=TEMP-(C(19)*X(5*J)-TWO*X(5*J-2))**2 -
     $              TEN*(X(5*J)+C(20)*X(5*J-4))**2 -
     $              (X(5*J-4)+X(5*J-2))**2
  80 CONTINUE
C
      F(8)=TEMP+1.0D3
C
      F(9)=-(C(21)*X(10)-TWO*X(11))**2 - X(10)*X(11) + 2.0D1
C
      F(10)=-(C(22)*X(50)-TEN*X(40))**2 - X(50)*X(40) + 5.0D1
C
      F(11)=-(C(23)*X(21)+X(22)+C(24)*X(23))**2 + 7.0D1
C
      RETURN
      END
```

APPENDIX B

MINIMIZING A QUADRATIC FUNCTION OVER A SET OF MINIMAL INFEASIBILITY

In this appendix, we solve the problem of minimizing a strictly convex quadratic function over a set of "minimally infeasible" points defined by a set of inconsistent linear inequalities. Let the original quadratic program be given by

$$(P) \quad \min_{x \in R^n} \quad (1/2)x'Qx + q'x$$

$$\text{subject to} \quad Ax \leq b,$$

where Q is a positive definite n x n matrix, q is an n x 1 vector, A is an m x n matrix, and b is an m x 1 vector. It can be shown that the following problem is dual to (P).

$$(D) \quad \min_{w \in R^m} \quad (1/2)w'Kw + k'w$$

$$\text{subject to} \quad w \geq 0,$$

where $K = AQ^{-1}A'$ and $k = AQ^{-1}q+b$. If (P) is feasible, then (P) has a solution x*, (D) has a solution w*, and

$$x^* = -Q^{-1}[A'w^* + q].$$

(See the discussion in section I.3.1.)

We now consider the case when (P) is infeasible. To analyze this case, we use the perturbed problem in (x, s):

$$(P_\varepsilon) \quad \min_{\substack{x \in R^n \\ s \in R^m}} \quad (1/2)x'Qx + q'x + \frac{1}{2\varepsilon} s's$$

$$\text{subject to} \quad Ax + s \leq b,$$

with $\varepsilon > 0$. Clearly, (P_ε) is always feasible. Note that (P_ε) is not defined at $\varepsilon = 0$, so that it is not clear that as $\varepsilon \to 0$ the solution of (P_ε) will be related to the solution of (P). Also note that s is not a slack variable since it is unconstrained in sign.

Given any x, it is necessary to choose s so that for each i, $i = 1, \ldots, m$:

$$(Ax - b)_i \leq 0 \to s_i = 0$$
$$(Ax - b)_i > 0 \to s_i = -(Ax - b)_i,$$

in order to minimize the objective function of (P_ε) over s. Thus we have

Prop. B1: (P_ε) is equivalent to the unconstrained problem

$$\min_x \quad (1/2)x'Qx + q'x + \frac{1}{2\varepsilon}(Ax - b)_+'(Ax - b)_+ \ .$$

Let

$$Z = \{x: \ |(Ax - b)_+|_2^2 \ \leq \ |(Az - b)_+|_2^2 \text{ for all } z \in R^n\},$$

i.e., Z is the set of points in R^n closest to feasibility in that the residual vector is smallest in norm. Note that Z is a convex set with $Z = \{x: Ax \leq b\}$ if the latter set is nonempty.

Example B1: $A = \begin{vmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{vmatrix}$ and $b = \begin{vmatrix} -1 \\ 0 \\ 0 \end{vmatrix}$

Define the constraints:

$$x_1 + x_2 \leq -1$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

which are clearly inconsistent. In this case, $Z = \{(-1/4, -1/4)\}$.

Example B2: $A = \begin{vmatrix} 1 & 1 \\ -1 & -1 \end{vmatrix}$ and $b = \begin{vmatrix} -1 \\ -2 \end{vmatrix}$

Define the constraints:

$$x_1 + x_2 \leq -1$$

$$x_1 + x_2 \geq 2$$

which are also inconsistent. In this case, $Z = \{x: x_1 + x_2 = 1/2\}$.

(P_ε) is a strictly convex program so that for every $\varepsilon > 0$ there exists a unique solution $(x^\varepsilon, s^\varepsilon)$ with

(B1) $s^\varepsilon = -(Ax^\varepsilon - b)_+$.

Let

$$\gamma = |(Ax - b)_+|_2^2 \text{ for } x \in Z \ (\gamma > 0),$$

$$\eta = \min_x \{(1/2)x'Qx + q'x\},$$

and

$$\zeta = \min_{x \in Z} (1/2)x'Qx + q'x \qquad (\zeta \geq \eta).$$

Denoting

$$Q_\varepsilon(x,s) = (1/2)x'Qx + q'x + \frac{1}{2\varepsilon}\, s's = Q(x) + \frac{1}{2\varepsilon}\, s's,$$

where $Q(x) = (1/2)x'Qx + q'x$, we have

$$\eta + \frac{1}{2\varepsilon}\,\gamma \leq Q_\varepsilon(x^\varepsilon,\, s^\varepsilon) \leq \zeta + \frac{1}{2\varepsilon}\,\gamma,$$

which implies $Q_\varepsilon(x^\varepsilon,\, s^\varepsilon) \to +\infty$ as $\varepsilon \to 0$ if and only if $\gamma > 0$. Thus

$$\text{(B2)}\quad \eta \leq Q_\varepsilon(x^\varepsilon,\, s^\varepsilon) - \frac{1}{2\varepsilon}\,\gamma = Q(x^\varepsilon) + \frac{1}{2\varepsilon}\,(s^{\varepsilon\prime}s^\varepsilon - \gamma) \leq \zeta.$$

Now it follows from (B1) that $s^{\varepsilon\prime}s^\varepsilon - \gamma > 0$. Therefore, since $|x^\varepsilon| \to \infty$ implies $Q(x^\varepsilon) \to +\infty$, (B2) implies that $\{x^\varepsilon\}$ must remain bounded and hence

$$\text{(B3)}\quad (s^{\varepsilon\prime}s^\varepsilon - \gamma) = O(\varepsilon).$$

We are led to the following result.

Prop. B2: Let $\{x^\varepsilon,\, s^\varepsilon)\}$ be the family of solutions to (P_ε). Then

$$\lim_{\varepsilon \to 0^+} (x^\varepsilon,\, s^\varepsilon) = (\hat{x}\,,\, \hat{s}\,)$$

where $\hat{x}$ is the unique solution to the problem

$$\min_{x} \quad (1/2)x'Qx + q'x$$

subject to $x \in Z$.

Pf: It follows from the earlier comments that $\{x^{\varepsilon}\}$ is bounded. Let $\hat{x}$ be any limit point of $\{x^{\varepsilon}\}$, i.e., there is a sequence $\{\varepsilon_\ell\} \to 0^+$ such that

$$\lim_{\ell \to \infty} x^{\varepsilon_\ell} = \hat{x}$$

It follows from (B1) that

$$\hat{s} = \lim_{\ell \to \infty} s^{\varepsilon_\ell} = -\lim_{\ell \to \infty} (Ax^{\varepsilon_\ell} - b)_+ = -(A\hat{x} - b)_+$$

exists and from (B3) that $\hat{s}'\hat{s} = \gamma$. Thus, $\hat{x} \in Z$. Now suppose, for contradiction, there is an $x^o \in Z$ such that

$$Q(\hat{x}) = (1/2)\hat{x}'Q\hat{x} + q'\hat{x} > (1/2)x^{o\prime}Qx^o + q'x^o = Q(x^o).$$

Let $s^o = -(Ax^o - b)_+$. Then, for any ε_ℓ

$$Q_{\varepsilon_\ell}(x^0, s^0) - Q_{\varepsilon_\ell}(x^{\varepsilon_\ell}, s^{\varepsilon_\ell}) =$$

$$Q(x^0) - Q(x^{\varepsilon_\ell}) + \frac{1}{2\varepsilon_\ell}(\gamma - |s^{\varepsilon_\ell}|_2^2) < 0$$

which contradicts the definition of $(x^{\varepsilon_\ell}, s^{\varepsilon_\ell})$.

If (P) is feasible, $(\hat{x}, \hat{s}) = (x^*, u^*)$, where u^* is the multiplier of the dual problem to (P) and x^* is the solution to (P). Proposition B2 is a special case of the penalty function theory for nonlinear programming.

Let $(x^\varepsilon, s^\varepsilon)$ be the solution to (P_ε) and let

$$J_\varepsilon = \{j: (Ax^\varepsilon - b)_j > 0\},$$

i.e., J_ε is the index set of violated constraints. Using the unconstrained form of (P_ε) (see Problem B1) we have that x^ε is a solution of

$$\min_x \quad (1/2)x'Qx + q'x + \frac{1}{2\varepsilon}|(Ax - b)_+|_2^2$$

if and only if

$$Qx^\varepsilon + q + \frac{1}{\varepsilon}\{\sum_{i\varepsilon J_\varepsilon} A_{i\cdot}(Ax^\varepsilon - b)_i\} = 0.$$

To construct a dual for (P_ε), we consider the Lagrangian function

$$L_\varepsilon(x, s, w) = (1/2)x'Qx + q'x + \frac{1}{2\varepsilon} s's + w'(Ax + s - b).$$

(P_ε) is equivalent to

$$\inf_{(x,s)} \sup_{w \geq 0} L_\varepsilon(x, s, w) = \inf_{(x,s)} \begin{cases} Q_\varepsilon(x, s): Ax + s \leq b \\ +\infty \quad : \text{otherwise} \end{cases}$$

and we define the dual problem to be

$$\sup_{w \geq 0} \inf_{(x,s)} L_\varepsilon(x, s, w).$$

For a given $w \geq 0$, we have that (x_w, s_w) minimizes $L_\varepsilon(x, s, w)$ if and only if

$$Qx_w + q + A'w = 0$$

and

$$\frac{1}{\varepsilon} s_w + w = 0.$$

Solving for x_w and s_w we obtain

$$x_w = -Q^{-1}[A'w + q]$$

and

$$s_w = -\varepsilon w.$$

Therefore,

$$L_\varepsilon(x_w, s_w, w) = (1/2)(A'w + q)'Q^{-1}QQ^{-1}(A'w + q)$$
$$-q'Q^{-1}(A'w + q) + \frac{1}{2\varepsilon}\varepsilon^2 w'w$$
$$+w'(-AQ^{-1}A'w - AQ^{-1}q - \varepsilon w - b)$$

$$= -(1/2)w'AQ^{-1}A'w - (AQ^{-1}q + b)'w - (1/2)\varepsilon w'w - (1/2)q'Q^{-1}q$$

and hence we have the following proposition.

Prop. B3: Solving the quadratic program

$$(D_\varepsilon) \quad \min_{w \in R^m} \quad (1/2)w'(K + \varepsilon I)w + k'w,$$

where

$$K = AQ^{-1}A'$$

and

$$k = AQ^{-1}q + b$$

is equivalent to solving (P_ε). Finally, we obtain the following theorem.

Theorem (Theorem 3 of Section I.3.1): Let $\{w^\varepsilon\}$ be the family of solutions to (D_ε) for positive values of ε and for each ε let x^ε be given by

$$x^\varepsilon = -Q^{-1}[A'w^\varepsilon + q].$$

Then

$$\lim_{\varepsilon \to 0^+} x^\varepsilon = \hat{x},$$

where $\hat{x}$ is a solution of the problem

$$\min_{x \in Z} \quad (1/2)x'Qx + q'x.$$

Pf: Propositions B1, B2, and B3 imply the theorem for problems having inequality constraints. Extension of the proof to include equality constraints is straightforward.

CNA PROFESSIONAL PAPERS - 1978 TO PRESENT*

PP 211
Mizrahi, Maurice M., "On Approximating the Circular Coverage Function," 14 pp., Feb 1978, AD A054 429

PP 212
Mangel, Marc, "On Singular Characteristic Initial Value Problems with Unique Solution," 20 pp., Jun 1978, AD A058 535

PP 213
Mangel, Marc, "Fluctuations in Systems with Multiple Steady States, Application to Lanchester Equations," 12 pp., Feb 78 (Presented at the First Annual Workshop on the Information Linkage Between Applied Mathematics and Industry, Naval PG School, 23-25 Feb 1978), AD A071 472

PP 214
Weinland, Robert G., "A Somewhat Different View of The Optimal Naval Posture," 37 pp., Jun 1978 (Presented at the 1976 Convention of the American Political Science Association (APSA/IUS Panel on "Changing Strategic Requirements and Military Posture"), Chicago, Ill., 2 Sep 1976), AD A056 228

PP 215
Colle, Russell C., "Comments on: Principles of Information Retrieval by Manfred Kochen," 10 pp., Mar 78 (Published as a Letter to the Editor, Journal of Documentation, Vol. 31, No. 4, Dec 1975), AD A054 426

PP 216
Colle, Russell C., "Lotka's Frequency Distribution of Scientific Productivity," 18 pp., Feb 1978 (Published in the Journal of the American Society for Information Science, Vol. 28, No. 6, Nov 1977), AD A054 425

PP 217
Colle, Russell C., "Bibliometric Studies of Scientific Productivity," 17 pp., Mar 78 (Presented at the Annual meeting of the American Society for Information Science held in San Francisco, California, Oct 1976), AD A054 442

PP 218 - Classified

PP 219
Huntzinger, R. LaVar, "Market Analysis with Rational Expectations: Theory and Estimation," 60 pp., Apr 78, AD A054 422

PP 220
Maurer, Donald E., "Diagonalization by Group Matrices," 26 pp., Apr 78, AD A054 443

PP 221
Weinland, Robert G., "Superpower Naval Diplomacy in the October 1973 Arab-Israeli War," 76 pp., Jun 1978 (Published in Seapower in the Mediterranean: Political Utility and Military Constraints, The Washington Papers No. 61, Beverly Hills and London: Sage Publications, 1979) AD A055 564

PP 222
Mizrahi, Maurice M., "Correspondence Rules and Path Integrals," 30 pp., Jun 1978 (Invited paper presented at the CNRS meeting on "Mathematical Problems in Feynman's Path Integrals," Marseille, France, 22-26 May 1978. Published in Springer Verlag Lecture Notes in Physics, 106, 1979), AD A055 536

PP 223
Mangel, Marc, "Stochastic Mechanics of Moleculeion Molecule Reactions," 21 pp., Jun 1978, AD A056 227

PP 224
Mangel, Marc, "Aggregation, Bifurcation, and Extinction in Exploited Animal Populations,"* 48 pp., Mar 1978, AD A058 536
*Portions of this work were completed at the Institute of Applied Mathematics and Statistics, University of British Columbia, Vancouver, B.C., Canada

PP 225
Mangel, Marc, "Oscillations, Fluctuations, and the Hopf Bifurcation,"* 43 pp., Jun 1978, AD A058 537
*Portions of this work were completed at the Institute of Applied Mathematics and Statistics, University of British Columbia, Vancouver, Canada.

PP 226
Ralston, J. M., and Mann, J. W.,* "Temperature and Current Dependence of Degradation in Red-Emitting GaP LEDs," 34 pp., Jun 1978 (Published in Journal of Applied Physics, Vol. 50, May 1979) AD A058 538
*Bell Telephone Laboratories, Inc.

PP 227
Mangel, Marc, "Uniform Treatment of Fluctuations at Critical Points," 50 pp., May 1978, AD A058 539

PP 228
Mangel, Marc, "Relaxation at Critical Points: Deterministic and Stochastic Theory," 54 pp., Jun 1978, AD A058 540

PP 229
Mangel, Marc, "Diffusion Theory of Reaction Rates, I: *Formulation and Einstein-Smoluchowski Approximation*," 50 pp., Jan 1978, AD A058 541

PP 230
Mangel, Marc, "Diffusion Theory of Reaction Rates, II Ornstein-Uhlenbeck Approximation," 34 pp., Feb 1978, AD A058 542

PP 231
Wilson, Desmond P., Jr., "Naval Projection Forces: The Case for a Responsive MAF," Aug 1978, AD A058 543

PP 232
Jacobson, Louis, "Can Policy Changes Be Made Acceptable to Labor?" 18 pp., Aug 1978 (Submitted for publication in Industrial and Labor Relations Review), AD A061 528

*CNA Professional Papers with an AD number may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151. Other papers are available from the Management Information Office, Center for Naval Analyses, 2000 North Beauregard Street, Alexandria, Virginia 22311. An Index of Selected Publications is also available on request. The Index includes a Listing of Professional Papers, with abstracts, issued from 1969 to June 1981.

PP 233
Jacobson, Louis, "An Alternative Explanation of the Cyclical Pattern of Quits," 23 pp., Nov 1978

PP 234 - Revised
Jondrow, James, and Levy, Robert A., "Does Federal Expenditure Displace State and Local Expenditure: The Case of Construction Grants," 25 pp., Oct 1979, AD A061 529

PP 235
Mizrahi, Maurice M., "The Semiclassical Expansion of the Anharmonic-Oscillator Propagator," 41 pp., Oct 1978 (Published in Journal of Mathematical Physics, Vol. 20, 1979), AD A061 538

PP 237
Maurer, Donald, "A Matrix Criterion for Normal Integral Bases," 10 pp., Jan 1979 (Published in the Illinois Journal of Mathematics, Vol. 22, 1978)

PP 238
Utgoff, Kathleen Classen, "Unemployment Insurance and The Employment Rate," 20 pp., Oct 1978 (Presented at the Conference on Economic Indicators and Performance: The Current Dilemma Facing Government and Business Leaders, presented by Indiana University Graduate School of Business), AD A061 527

PP 239
Trost, R. P., and Warner, J. T., "The Effects of Military Occupational Training on Civilian Earnings: An Income Selectivity Approach," 38 pp., Nov 1979, AD A077 831

PP 240
Powers, Bruce, "Goals of the Center for Naval Analyses," 13 pp., Dec 1978, AD A063 759

PP 241
Mangel, Marc, "Fluctuations at Chemical Instabilities," 12 pp., Dec 1978 (Published in Journal of Chemical Physics, Vol. 69, No. 8, 15 Oct 1978), AD A063 787

PP 242
Simpson, William R., "The Analysis of Dynamically Interactive Systems (Air Combat by the Numbers)," 160 pp., Dec 1978, AD A063 760

PP 243
Simpson, William R., "A Probabilistic Formulation of Murphy Dynamics as Applied to the Analysis of Operational Research Problems," 18 pp., Dec 1978, AD A063 761

PP 244
Sherman, Allan, and Horowitz, Stanley A., "Maintenance Costs of Complex Equipment," 20 pp., Dec 1978 (Published By The American Society of Naval Engineers, Naval Engineers Journal, Vol. 91, No. 6, Dec 1979), AD A071 473

PP 245
Simpson, William R., "The Accelerometer Methods of Obtaining Aircraft Performance from Flight Test Data (Dynamic Performance Testing)," 403 pp., Jun 1979, AD A075 226

PP 246
Brechling, Frank, "Layoffs and Unemployment Insurance," 35 pp., Feb 1979 (Presented at the NBER Conference on "Low Income Labor Markets," Chicago, Jun 1978), AD A096 629

PP 248
Thomas, James A., Jr., "The Transport Properties of Dilute Gases in Applied Fields," 183 pp., Mar 1979, AD A096 464

PP 249
Glasser, Kenneth S., "A Secretary Problem with a Random Number of Choices," 23 pp., Mar 1979

PP 250
Mangel, Marc, "Modeling Fluctuations in Macroscopic Systems," 26 pp., Jun 1979

PP 251
Trost, Robert P., "The Estimation and Interpretation of Several Selectivity Models," 37 pp., Jun 1979, AD A075 941

PP 252
Nunn, Walter R., "Position Finding with Prior Knowledge of Covariance Parameters," 5 pp., Jun 1979 (Published in IEEE Transactions on Aerospace & Electronic Systems, Vol. AES-15, No. 3, Mar 1979)

PP 253
Glasser, Kenneth S., "The d-Choice Secretary Problem," 32 pp., Jun 1979, AD A075 225

PP 254
Mangel, Marc, and Quanbeck, David B., "Integration of a Bivariate Normal Over an Offset Circle," 14 pp., Jun 1979, AD A096 471

PP 255 - Classified, AD B051 441L

PP 256
Maurer, Donald E., "Using Personnel Distribution Models," 27 pp., Feb 1980, AD A082 218

PP 257
Thaler, R., "Discounting and Fiscal Constraints: Why Discounting is Always Right," 10 pp., Aug 1979, AD A075 224

PP 258
Mangel, Marc S., and Thomas, James A., Jr., "Analytical Methods in Search Theory," 86 pp., Nov 1979, AD A077 832

PP 259
Glass, David V.; Hsu, Ih-Ching; Nunn, Walter R.; and Perin, David A., "An Analysis of a Layered Defense Model," 17 pp., Mar 1980, AD A077 833

PP 260
Mangel, Marc S., and Cope, Davis K., "Detection Rate and Sweep Width in Visual Search," 14 pp., Nov 1979, AD A077 834

PP 261
Vila, Carlos L.; Zviljec, David J; and Ross, John, "Franck-Condon Theory of Chemical Dynamics. VI. Angular Distributions of Reaction Products," 14 pp., Nov 1979 (Reprinted from Journal Chemical Phys. 70(12), 15 Jun 1979), AD A076 287

PP 262
Petersen, Charles C., "Third World Military Elites in Soviet Perspective," 50 pp., Nov 1979, AD A077 835

PP 263
Robinson, Kathy I., "Using Commercial Tankers and Containerships for Navy Underway Replenishment," 25 pp., Nov 1979, AD A077 836

PP 264
Weinland, Robert G., "The U.S. Navy in the Pacific: Past, Present, and Glimpses of the Future," 31 pp., Nov 1979 (Delivered at the International Symposium on the Sea, sponsored by the International Institute for Strategic Studies, The Brookings Institution and the Yomiuri Shimbun, Tokyo, 16-20 Oct 1978), AD A077 837

PP 265
Weinland, Robert G., "War and Peace in the North: Some Political Implications of the Changing Military Situation in Northern Europe," 18 pp., Nov 1979 (Prepared for presentation to the Conference of the Nordic Balance in Perspective: The Changing Military and Political Situation," Center for Strategic and International Studies, Georgetown University, 15-16 Jun 1978), AD A077 838

PP 266
Utgoff, Kathleen Classen, and Brechling, Frank, "Taxes and Inflation," 25 pp., Sep 1979, AD A081 194

PP 267
Trost, Robert P., and Vogel, Robert C., "The Response of State Government Receipts to Economic Fluctuations and the Allocation of Counter-Cyclical Revenue Sharing Grants," 12 pp., Dec 1979 (Reprinted from the Review of Economics and Statistics, Vol. LXI, No. 3, Aug 1979)

PP 268
Thomason, James S., "Seaport Dependence and Inter-State Cooperation: The Case of Sub-Saharan Africa," 141 pp., Jan 1980, AD A081 193

PP 269
Weiss, Kenneth G., "The Soviet Involvement in the Ogaden War," 42 pp., Jan 1980 (Presented at the Southern Conference on Slavic Studies in October, 1979), AD A082 219

PP 270
Remnek, Richard, "Soviet Policy in the Horn of Africa: The Decision to Intervene," 52 pp., Jan 1980 (To be published in "The Soviet Union in the Third World: Success or Failure," ed. by Robert H. Donaldson, Westview Press, Boulder, Co., Summer 1980), AD A081 195

PP 271
McConnell, James, "Soviet and American Strategic Doctrines: One More Time," 43 pp., Jan 1980, AD A081 192

PP 272
Weiss, Kenneth G., "The Azores in Diplomacy and Strategy, 1940-1945," 46 pp., Mar 1980, AD A085 094

PP 273
Nakada, Michael K., "Labor Supply of Wives with Husbands Employed Either Full-Time or Part-Time," 39 pp., Mar 1980, AD A082 220

PP 274
Nunn, Walter R., "A Result in the Theory of Spiral Search," 9 pp., Mar 1980, AD A112 481

PP 275
Goldberg, Lawrence, "Recruiters, Advertising, and Navy Enlistments," 34 pp., Mar 1980, AD A082 221

PP 276
Goldberg, Lawrence, "Delaying an Overhaul and Ship's Equipment," 40 pp., May 1980, AD A085 095

PP 277
Mangel, Marc, "Small Fluctuations in Systems with Multiple Limit Cycles," 19 pp., Mar 1979 (Published in SIAM J. Appl. Math., Vol. 38, No. 2, Feb 1980), AD A086 229

PP 278
Mizrahi, Maurice, "A Targeting Problem: Exact vs. Expected-Value Approaches," 23 pp., Apr 1980, AD A085 096

PP 279
Walt, Stephen M., "Causal Inferences and the Use of Force: A Critique of Force Without War," 50 pp., May 1980, AD A085 097

PP 280
Goldberg, Lawrence, "Estimation of the Effects of a Ship's Steaming on the Failure Rate of its Equipment: An Application of Econometric Analysis," 25 pp., Apr 1980, AD A085 098

PP 281
Mizrahi, Maurice M., "Comment on 'Discretization Problems of Functional Integrals in Phase Space'," 2 pp., May 1980, (Published in "Physical Review D", Vol. 22, No. 8, 15 Oct 1980), AD A094 994

PP 283
Dismukes, Bradford, "Expected Demand for the U.S. Navy to Serve as An Instrument of U.S. Foreign Policy: Thinking About Political and Military Environmental Factors," 30 pp., Apr 1980, AD A085 099

PP 284
Keilson, J.;* Nunn, W.; and Sumita, U.,** "The Laguerre Transform," 119 pp., May 1930, AD A085 100
*The Graduate School of Management, University of Rochester and the Center for Naval Analyses
**The Graduate School of Management, University of Rochester

PP 285
Remnek, Richard B., "Superpower Security Interests in the Indian Ocean Area," 26 pp., Jun 1980, AD A087 113

PP 286
Mizrahi, Maurice M., "On the WKB Approximation to the Propagator for Arbitrary Hamiltonians," 25 pp., Aug 1980 (Published in Journal of Math. Phys., 22(1) Jan 1981), AD A091 307

PP 287
Cope, Davis, "Limit Cycle Solutions of Reaction-Diffusion Equations," 319 pp., Jun 1980, AD A087 114

PP 288
Golman, Walter, "Don't Let Your Slides Flip You: A Painless Guide to Visuals That Really Aid," 28 pp., (revised Aug 1982), AD A092 732

PP 289
Robinson, Jack, "Adequate Classification Guidance - A Solution and a Problem," 7 pp., Aug 1980, AD A091 212

PP 290
Watson, Gregory H., "Evaluation of Computer Software in an Operational Environment," 17 pp., Aug 1980, AD A091 213

PP 291
Maddala, G. S.* and Trost, R. P., "Some Extensions of the Nerlov Press Model," 17 pp., Oct 1980, AD A091 946
*University of Florida

PP 292
Thomas, James A., Jr., "The Transport Properties of Binary Gas Mixtures in Applied Magnetic Fields,: 10 pp., Sep 1980 (Published in Journal of Chemical Physics 72(10), 15 May 1980)

PP 293
Thomas, James A., Jr., "Evaluation of Kinetic Theory Collision Integrals Using the Generalized Phase Shift Approach," 12 pp., Sep 1980 (Printed in Journal of Chemical Physics 72(10), 15 May 1980)

PP 294
Roberts, Stephen S., "French Naval Policy Outside of Europe," 30 pp., Sep 1980 (Presented at the Conference of the Section on Military Studies, International Studies Association, Kiawah Island, S.C.), AD A091 306

PP 295
Roberts, Stephen S., "An Indicator of Informal Empire: Patterns of U.S. Navy Cruising on Oversees Stations, 1869-1897," 40 pp., Sep 1980 (Presented at Fourth Naval History Symposium, US Naval Academy, 26 Oct 1979), AD A091 316

PP 296
Dismukes, Bradford, and Petersen, Charles C., "Maritime Factors Affecting Iberian Security," (Factores Maritimos que Afectan la Seguridad Iberica) 14 pp., Oct 1980, AD A092 733

PP 297 - Classified

PP 298
Mizrahi, Maurice M., "A Markov Approach to Large Missile Attacks," 31 pp., Jan 1981, AD A096,159

PP 299
Jondrow, James M., and Levy, Robert A., "Wage Leadership in Construction, 19 pp., Jan 1981, AD A094 797

PP 300
Jondrow, James, and Schmidt, Peter,* "On the Estimation of Technical Inefficiency in the Stochastic Frontier Production Function Model," 11 pp., Jan 1981, AD A096 160
*Michigan State University

PP 301
Jondrow, James M.; Levy, Robert A.; and Hughes, Claire, "Technical Change and Employment in Steel, Autos, Aluminum, and Iron Ore," 17 pp., Mar 1981, AD A099 394

PP 302
Jondrow, James M., and Levy, Robert A., "The Effect of Imports on Employment Under Rational Expectations," 19 pp., Apr 1981, AD A099 392

PP 303
Thomason, James, "The Rarest Commodity in the Coming Resource Wars," 3 pp., Aug 1981 (Published in the Washington Star, 13 Apr 1981), AD A104 221

PP 304
Duffy, Michael K.; Greenwood, Michael J.;* and McDowell, John M.,** "A Cross-Sectional Model of Annual Interregional Migration and Employment Growth: Intertemporal Evidence of Structural Change, 1958-1975," 31 pp., Apr 1981, AD A099 393
*University of Colorado
**Arizona State University

PP 305
Nunn, Laura H., "An Introduction to the Literature of Search Theory," 32 pp., Jun 1981, AD A100 420

PP 306
Anger, Thomas E., "What Good Are Warfare Models?" 7 pp., May 1981, AD A100 421

PP 307
Thomason, James, "Dependence, Risk, and Vulnerability," 43 pp., Jun 1981, AD A102 698

PP 308
Mizrahi, M.M., "Correspondence Rules and Path Integrals, 17 pp., Jul 1981. (Published in "Nuovo Cimento B", Vol. 61, 1981), AD A102 699

PP 309
Weinland, Robert G., "An (The?) Explanation of the Soviet Invasion of Afghanistan," 44 pp., May 1981, AD A100 422

PP 310
Stanford, Janette M., and Wu, Tai Te,* "A Predictive Method for Determining Possible Three-dimensional Foldings of Immunoglobulin Backbones Around Antibody Combining Sites," 19 pp., Jun 1981 (Published in J. Theor. Biol., 1981), 88, AD A100 423
*Northwestern University

PP 311
Bowes, Marianne, Brechling; Frank P. R.; and Utgoff, Kathleen P. Classen, "An Evaluation of UI Funds," 13 pp., May 1981 (Published in National Commission on Unemployment Compensation's "Unemployment Compensation: Studies and Research," Volume 2, Jul 1980), AD A100 424

PP 312
Jondrow, James; Bowes, Marianne; and Levy, Robert, "The Optimum Speed Limit," 23 pp., Jul 1983 (Revised), AD A100 425

PP 313
Roberts, Stephen S., "The U.S. Navy in the 1980s," 36 pp., Jul 1981, AD A102 696

PP 314
Jehn, Christopher; Horowitz, Stanley A.; and Lockman, Robert F., "Examining the Draft Debate," 20 pp., Jul 1981, AD A106 192

PP 315
Buck, Ralph V. (Capt., USN), "Le Catastrophe by any other name...," 4 pp., Jul 1981, AD A102 697

PP 316
Roberts, Stephen S., "Western European and NATO Navies, 1980," 20 pp., Aug 1981, AD A104 223

PP 317
Roberts, Stephen S., "Superpower Naval Crisis Management in the Mediterranean," 35 pp., Aug 1981, AD A104 222

PP 318
Vego, Milan N., "Yugoslavia and the Soviet Policy of Force in the Mediterranean Since 1961," 187 pp., Aug 1981

PP 319
Smith, Michael W., "Antiair Warfare Defense of Ships at Sea," 46 pp., Sep 1981 (This talk was delivered at the Naval Warfare System and Technology Conference of the American Institute of Aeronautics and Astronautics in Washington on 12 Dec 1980; in Boston on 20 Jan 1981; and in Los Angeles on 12 Jun 1981.), AD A106 191

PP 320
Trost, R. P.; Lurie, Philip; and Berger, Edward, "A Note on Estimating Continuous Time Decision Models," 15 pp., Sep 1981, AD A106 193

PP 321
Duffy, Michael K., and Ladman, Jerry R.,* "The Simultaneous Determination of Income and Employment in United States--Mexico Border Region Economies," 34 pp., Sep 1981, AD A106 540
*Associate Professor of Economics, Arizona State University

PP 322
Warner, John T., "Issues in Navy Manpower Research and Policy: An Economist's Perspective," 66 pp., Dec 1981, AD A110 221

PP 323
Bomse, Frederick M., "Generation of Correlated Log-Normal Sequences for the Simulation of Clutter Echoes," 33 pp., Dec 1981

PP 324
Horowitz, Stanley A., "Quantifying Seapower Readiness," 6 pp., Dec 1981 (Published in Defense Management Journal, Vol. 18, No. 2), AD A110 220

PP 326
Roberts, Stephen S., "Western European and NATO Navies, 1981," 27 pp., Jul 1982, AD A118 703

PP 327
Hammon, Colin (Capt., USN), and Graham, David R., "Estimation and Analysis of Navy Shipbuilding Program Disruption Costs," 12 pp., Mar 1980, AD A112 514

PP 328
Weinland, Robert G., "Northern Waters: Their Strategic Significance," 27 pp., Dec 1980, AD A112 509

PP 329
Mangel, Marc, "Applied Mathematicians And Naval Operators," 40 pp., Mar 1982 (Revised), AD A116 598

PP 330
Lockman, Robert F., "Alternative Approaches to Attrition Management," 30 pp., Jan 1982, AD A112 510

PP 331
Roberts, Stephen S., "The Turkish Straits and the Soviet Navy in the Mediterranean," 15 pp., Mar 1982 (Published in Navy International)

PP 332
John, Christopher, "The RDF and Amphibious Warfare," 36 pp., Mar 1982, AD A113 592

PP 333
Lee, Lung-Fei,* and Trost, Robert P., "Estimation of Some Limited Dependent Variable Models with Application to Housing Demand," 26 pp., Jan 1982. (Published in Journal of Econometrics 8 (1978), AD A 112 536
*University of Minnesota

PP 334
Kenny, Lawrence W.;* Lee, Lung-Fei;** Maddala, G. S.;* and Trost R. P., "Returns to College Education: An Investigation of Self-Selection Bias Based on the Project Talent Data," 15 pp., Jan 1982. (Published in International Economic Review, Vol. 20, No. 3, Oct 1979), AD A112 480
*University of Florida
**University of Minnesota

PP 335
Lee, Lung-Fei;* Maddala, G. S.;** and Trost, R.P. "Asymptotic Covariance Matrices of Two-Stage Probit and Two-Stage Tobit Methods for Simultaneous Equations Models with Selectivity," 13 pp., Jan 1982. (Published in Econometrica, Vol. 48, No. 2, Mar 1980), AD A112 483
*University of Minnesota
**University of Florida

PP 336
O'Neill, Thomas, "Mobility Fuels for the Navy," 13 pp., Jan 1982. (Accepted for publication in Naval Institute Proceedings), AD A112 511

PP 337
Warner, John T., and Goldberg, Matthew S., "The Influence of Non-Pecuniary Factors on Labor Supply: The Case of Navy Enlistment Personnel," 23 pp., Dec 1981, AD A113 094

PP 338
Feldman, Paul and Jondrow, James, "American Journal of Political Science," 19 pp., Feb 1984

PP 339
Wilson, Desmond P., "The Persian Gulf and the National Interest," 11 pp., Feb 1982, AD A112 505

PP 340
Lurie, Philip; Trost, R. P.; and Berger, Edward, "A Method for Analyzing Multiple Spell Duration Data," 34 pp., Feb 1982, AD A112 504

PP 341
Trost, Robert P., and Vogel, Robert C.,* "Prediction with Pooled Cross-Section and Time-Series Data: Two Case Studies," 6 pp., Feb 1982, AD A112 503
*Southern Illinois University

PP 342
Lee, Lung-Fei;* Maddala, G. S.;** and Trost, R. P., "Testing for Structural Change by D-Methods in Switching Simultaneous Equations Models," 5 pp., Feb 1982, AD A112 482
*University of Minnesota
**University of Florida

PP 343
Goldberg, Matthew S., "Projecting the Navy Enlisted Force Level," 9 pp., Feb 1982, AD A112 484

PP 344
Fletcher, Jean, W., "Navy Quality of Life and Reenlistment," 13 pp., Nov 1981, AD A113 095

PP 345
Utgoff, Kathy, and Thaler, Dick,* "The Economics of Multi Year Contracting," 47 pp., Mar 1982. (Presented at the 1982 Annual Meeting of the Public Choice Society, San Antonio, Texas, 5-7 Mar 1982), AD A114 732
*Cornell University

PP 346
Rostker, Bernard, "Selective Service and the All-Volunteer Force," 23 pp., Mar 1982, AD A113 096

PP 347
McConnell, James, M., "A Possible Counterforce Role for the Typhoon," 24 pp., Mar 1982, AD A116 601

PP 348
Jondrow, James, and Trost, Robert, "An Empirical Study of Production Inefficiency in the Presence of Errors-in-The-Variables," 14 pp., Feb 1982, AD A113 591

PP 349
Breckenridge, W. H.,* and Maimin, O. Kim, "Collisional Intramultiplet Relaxation of $Cd(5s5p^3P_{0,1,2})$ by Alkane Hydrocarbons," 7 pp., Jul 1981. (Published in Journal of Chemical Physics, 76(4), 15 Feb 1982), AD A113 093
*University of Utah, Dept. of Chemistry

PP 350
Levin, Marc, "A Method for Increasing the Firepower of Virginia Class Cruisers," 10 pp., Apr 1982. (To be published in U.S. Naval Institute Proceedings), AD A116 602

PP 351
Coutre, S. E.;* Stanford, J. M.; Hovis, J. G.;* Stevens, P. W.;* and Wu, T. T.,* "Possible Three-Dimensional Backbone Folding Around Antibody Combining Site of Immunoglobulin MOPC 167," 18 pp., Apr 1982 (Published in Journal of Theoretical Biology)
*Northwestern University, Depts. of Biochemistry & Molecular Biology and Engineering Sciences & Applied Mathematics

PP 352
Barfoot, C. Bernard, "Aggregation of Conditional Absorbing Markov Chains," 7 pp., Jun 1982 (Presented to the Sixth European Meeting on Cybernetics and Systems Research, held at the University of Vienna, Apr 1982), AD A116 603

PP 353
Barfoot, C. Bernard, "Some Mathematical Methods for Modeling the Performance of a Distributed Data Base System," 18 pp., Jun 1982. (Presented to the International Working Conference on Model Realism, held at Bad Honnek, West Germany, Apr 1982), AD A116 604

PP 354
Hall, John V., "Why the Short-War Scenario is Wrong for Naval Planning," 6 pp., Jun 1982., AD A118 702

PP 356
Cylke, Steven; Goldberg, Matthew S.; Hogan, Paul; and Mairs, Lee; "Estimation of the Personal Discount Rate: Evidence from Military Reenlistment Decisions," 19 pp., Apr 1982, AD A122 419

PP 357
Goldberg, Matthew S., "Discrimination, Nepotism, and Long-Run Wage Differentials," 13 pp., Sep 1982 (Published in Quarterly Journal of Economics, May 1982)

PP 358
Akst, George, "Evaluating Tactical Command And Control Systems--A Three-Tiered Approach," 12 pp., Sep 1982, AD 122 478

PP 359
Quester, Aline; Fletcher, Jean; and Marcus, Alan; "Veteran Status as a Screening Device: Comment," 26 pp., Aug 1982, AD A123 658

PP 361
Quanbeck, David B., "Methods for Generating Aircraft Trajectories," 51 pp., Sep 1982, AD A122 386

PP 362
Horowitz, Stanley A., "Is the Military Budget Out of Balance?," 10 pp., Sep 1982, AD A122 368

PP 363
Marcus, A. J., "Personnel Substitution and Navy Aviation Readiness," 35 pp., Oct 1982, AD A122 420

PP 364
Quester, Aline, and Nakada, Michael, "The Military's Monopsony Power," 29 pp., Oct 1982, AD A123 657

PP 365
Greer, William L., and Bartholomew, James C., (Cdr,USN), Psycychological Aspects of Mine Warfare," 15 pp., Oct 1982 AD A128 244

PP 366
Spruill, Nancy L., and Gastwirth, Joseph L.,* "On the Estimation of the Correlation Coefficient From Grouped Data," 9 pp., Oct 1982, (Published in the Journal of the American Statistical Association, Sep 1982, Vol. 77, No. 379, Theory and Methods Section), AD A122 382
*George Washington University, Dept. of Statistics

PP 367
Petersen, Charles C., "Soviet Tactics for Warfare at Sea (Two Decades of Upheaval)," 57 pp., Nov 1982

PP 368
Weinland, Robert G., "The Evolution of Soviet Requirements for Naval Forces--Solving the Problems of the Early 1960s," 41 pp., Dec 1982, AD A123 655

PP 369
Quester, Aline, and Lockman, Robert, "The All-Volunteer Force: A Positive Perspective," 29 pp., Nov 1982, AD A128 279

PP 370
Rostker, Bernard D., "Human Resource Models: An Overview," 17 pp., Nov 1982, AD A123 656

PP 372
Hurley, William J., "An Overview of Acoustic Analysis," 46 pp., Jan 1983, AD A128 316

PP 373
Jacobson, Louis, "Research to Quantify the Effect of Permanent Change of Station Moves on Wives' Wages and Labor Supply," 35 pp., Jan 1983, AD A128 300

PP 374
Clay-Mendez, Deborah, and Balis, Ellen, "Balancing Accession and Retention: The Disaggregate Model," 27 pp., Aug 1982

PP 375
Feldman, Paul, "Privatizing Airports in Washington, D.C.," 17 pp., Feb. 1983, AD A128 236

PP 376
Weiss, Kenneth G., "Power Grows Out of the Barrel of a Gunboat: The U.S. in Sino-Soviet Crises," 136 pp., Dec 1982

PP 377
Anger, Thomas E., "The Outlook for Military Operations Research," 14 pp., Apr 1983

PP 379
Jondrow, James M.; Chase, David E.; and Gamble, Christopher L., "The Price Differential Between Domestic and Imported Steel," 17 pp., May 1983

PP 380
Balis, Ellen, "Balancing Accession and Retention: Cost and Productivity Tradeoffs," 38 pp., Mar 1983

PP 381
Reeves, John M. L., "CNA's Conceptual Design and Cost Models for High-Speed Surface Craft," 23 pp., Apr 1983, AD A128 245

PP 382
Levy, Robert A., and Jondrow, James M, "The Adjustment of Employment to Technical Change in the Steel and Auto Industries," 40 pp., May 1983

PP 383 (Revised)
Thomas, James A., Jr, and Mangel, Marc, "Properties of Quick Look Passive Localization," 39 pp., Jul 1983

PP 384
Goldberg, Matthew S., and Hager, Michael F., "A Comparison of the Prophet and ACOL Force Projection Models, " 35 pp., Jun 1981

PP 385
Angier, Bruce; Driscoll, Kurt; and Gregory, David, "Manpower Requirements Derivation for the Navy Comprehensive Compensation and Supply Study," 22 pp., Sep 1982

PP 386
Angier, Bruce N.; Driscoll, Kurt A.; and Carpenter, Kathy A., "Construction of 'Training Cost Per Graduate' for the Navy Comprehensive Compensation and Supply Study," 67 pp., Nov 1982

PP 387
Balis, Ellen, and Clay-Mendez, Deborah, "Balancing Accession and Retention: The Aggregate Model," 20 pp., Jul 1982

PP 388
Clay-Mendez, Deborah, "Models of Accession and Retention," 11 pp., Oct 1982

PP 389
Clay-Mendez, Deborah, "A Minimum Recruiting Cost Function for Male High School Graduates," 31 pp., Jan 1982

PP 390
Clay-Mendez, Deborah, "Documentation for the Recruiting Cost Estimates Utilized in the Navy Comprehensive Compensation and Supply Study," 30 pp., Sep 1982

PP 391
Goldberg, Larry, "Summary of Navy Enlisted Supply Study," 11 pp., Jul 1981

PP 392
Warner, John T., and Simon, Bruce, "An Empirical Analysis of Pay and Navy Enlisted Retention in the AVF: Preliminary Results," 51 pp., Dec 1979

PP 394
McGibney, Donald; Camerini, Ugo; Roberts, Arthur; and Winston, Roland, "Development of an Underwater High Sensitivity Cherenkov Detector: Sea Urchin," 20 pp., Aug 1983

PP 395
Curran, Lawrence and Quester, Aline O., "Retention in the Navy's Selected Reserve: An Analysis Combining Survey and Personnel Data Records," 17 pp., Jun 1984

PP 396
Jondrow, James M.; Brechling, Frank; and Marcus, Alan; "Older Workers in the Market for Part-Time Employment," 34 pp., Aug 1983

PP 398
Levy, Robert A.; Bowes, Marianne; and Jondrow, James M.; "Technical Change and Employment in the Steel, Auto, Aluminum, Coal, and Iron Ore Industries," 25 pp., Sep 1983

PP 399
Roberts, Stephen, "Western European and NATO Navies," 23 pp., Nov 1982

PP 400
Laird, Robbin F., "French Nuclear Forces in the 1980s and 1990s," 37 pp., Aug 1983

PP 401
Maloney, Arthur P., "The Berlin-Baghdad Railway as a Cause of World War I," 27 pp., Jan 1984

PP 402
Herz, Henry L., "A Parametric Analysis of Duels," 41 pp., Mar 1984

PP 405
Petersen, Charles C., "Aircraft Carriers in Soviet Naval Theory From 1960 to the Falklands War," 20 pp., Jan 1984

PP 406
Bowes, Marianne, "Profit-Maximizing vs. Optimal Behavior in a Spatial Setting: Summary and Extensions," 10 pp., Jan 1984

PP 407
Laird, Robbin F., "The French Strategic Dilemma," 41 pp., Mar 1984

PP 408
Marcus, Alan, and Quester, Aline, "Measuring the Productivity of First Term Navy Enlistees," 30 pp., Apr 1984

PP 410
Weinland, Robert G., "Soviet Strategy and the Objectives of Their Naval Presence in the Mediterranean," 44 pp., May 1984 (Published as Giacomo, Luciani (ed.), The Mediterranean Region: Economic Interdependence and the Future of Society, (London & Canberra/New York: Croom-Helm/St. Martin's Press 1984, pp. 267-291)

PP 411
Horowitz, Stan, "Skill Mix, Experience and Readiness," 12 pp., Oct 1983

PP 412
McConnell, James M., "The Interacting Evolution of Soviet and American Military Doctrines," 119 pp., Sep 1980

PP 413
Quester, Aline O. and Sicilia, Anne S., "Naval Reserve Forces Throughout the World," 25 pp., Jun 1984

PP 414
Klotz, Ambry, "Career Development of Managers and Executives: A Compedium," 38 pp., Jun 1984

PP 414
Mizrahi, Maurice M., "Ca Authoritative Studies Be Trusted?" 2 pp., Jun 1984

PP 416
Nickel, Ronald H. and Tolle, Jon W., "A Sequential Quadratic Programming Algorithm For Solving Large, Sparse Nonlinear Programs," 143 pp., May 1984